SPECIALE COMUNICAZIONE

ANNO 3 - N. 12
MAGGIO 1990

L. 14.000
Frs. 21.00

IN COLLABORAZIONE CON AMIGA

MAGAZINE AMIGA

IL MENSILE JACKSON PER GLI UTENTI DI AMIGA

- Personalizzare Amiga
- Il tempo di Amiga
- Music-X (Parte II)
- Parola ad Amiga
- Amiga Tex: novità DTP
- Uomini, topi e computer
- Audio Video Digitizer

INSERTO LE PAGINE DI Amiga Transactor PER EDIZIONE ITALIANA

Spedizione in abbonamento postale Gruppo III/70 Taxe perçue (Tassa riscossa) Milano Ferr. Corr.

GRUPPO EDITORIALE JACKSON

AMIGA MAGAZINE
ANNO 3 - NUMERO 12
MAGGIO 1990

Sommario

GENIAS, PER AMPLIARE L'ORGANICO DI COLLABORATORI, CERCA **PROGRAMMATORI, GRAFICI E MUSICISTI** DA INSERIRE NEI PROPRI TEAM. **GENIAS** E' LA PRIMA (E FINORA) UNICA SOFTWARE-HOUSE ITALIANA A DISTRIBUIRE VIDEOGIOCHI IN TUTTA EUROPA E PROSSIMAMENTE NEL MONDO.

VUOI REALIZZARE UN VIDEO GAME, DI SICURO SUCCESSO, SU UNA DELLE SEGUENTI MACCHINE?

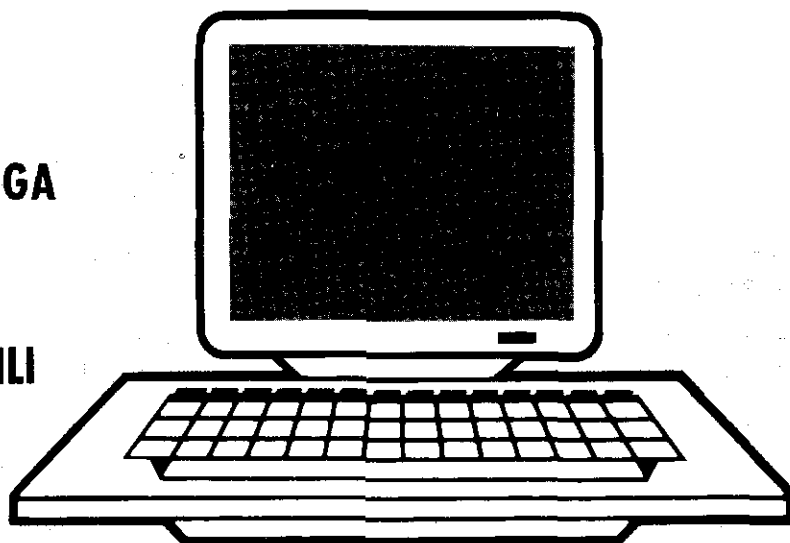
COMMODORE 64

COMMODORE AMIGA

ATARI ST

MS-DOS COMPATIBILI

SPECTRUM

AMSTRAD

SE LA RISPOSTA E' **SI**, ALLORA SCRIVICI, ALL'INDIRIZZO SOTTO INDICATO INVIANDO POSSIBILMENTE UN PICCOLO DEMO DELLE TUE CAPACITA'.

GENIAS Via Bentivogli, 55 - 40055 Castenaso (BO)

Direttore Responsabile: Paolo Reina
Coordinamento Tecnico e Redazionale: Massimiliano Anticoli
Tel. 02/6948287
Redazione: Stefano Albarelli (On Disk), Romano Tenca (Transactor)
Segreteria di redazione: Elena Ferré - Tel. 02/6948254
Art Director: Marcello Longhini
Copertina, Grafica e Impaginazione elettronica: Cristina Turra
Collaboratori: Lucio Bragagnolo, Cesare Palmieri, Alfonso Romano, Stefano Riva, Sergio Ruocco, Sebastiano Vigna, Daniele Cassanelli, Mauro Ballocchi, Aldo e Andrea Laus, Mariano Martin Gianni Biagini, Renato Acciardi, Stefano Paganini

Group Publisher: Pierantonio Palerma
Direzione Coordinamento Operativo: Graziella Falaguasta
Publisher Area Consumer: Filippo Canavese
Pubblicità: Filippo Canavese
Direzione Sviluppo Pubblicità: Walter Bussolera

SEDE LEGALE
Via P. Mascagni, 14 - 20122 Milano

DIREZIONE - REDAZIONE
Via Pola, 9 - 20124 Milano - Tel.: 02/69481
Fax: 02/6948238 Telex 316213 REINA I

PUBBLICITA'
Via Pola, 9 - 20124 Milano - Tel.: 02/6948254
ROMA - LAZIO E CENTRO SUD Via Lago di Tana, 16
00199 Roma
Tel.: 06/8380547 - Fax: 06/8380637

INTERNATIONAL MARKETING
Tel.: 02/6948233

DIREZIONE AMMINISTRATIVA
Via Rosellini, 12 - 20124 Milano Tel.: 02/69481
Fax: 02/6948238

UFFICIO ABBONAMENTI
Via Rosellini, 12 - 20124 Milano - Fax: 02/6948489
Telex 333436GEJ IT - Tel.: 02/6948490 (nei giorni di martedì, mercoledì, giovedì 14.30 - 17.30)

Prezzo della rivista: L. 14.000 prezzo arretrato L. 28.000
Abbonamento annuo Italia L. 124.000, Estero L. 248.000
I versamenti vanno indirizzati a:
Gruppo Editoriale Jackson SpA
Via Rosellini, 12 - 20124 Milano, mediante l'emissione di assegno bancario o per contanti. L'abbonamento può essere sottoscritto anche utilizzando il c/c postale 11666203

CONSOCIATE ESTERE
GEJ Publishing Group Inc. Los Altos Hills
27910 Roble Blanco
94022 California - Tel.: (001 - 415 - 9492028)
Grupo Editorial Jackson - Calle Alcantara, 57
28006 Madrid - Tel.: 1/4017365

Stampa: F.B.M. (Gorgonzola)
Fotolito: Foligraph (Milano)
Distribuzione: Sodip - Via Zuretti, 25 - 20125 Milano

Il Gruppo Editoriale Jackson è iscritto al Registro Nazionale della stampa al N. 117 Vol. 2 foglio 129 in data 17/8/1982.
Spedizione in abbonamento postale gruppo III/70
Aut. Trib. di Milano n. 102 del 22/2/1988

Il Gruppo Editoriale Jackson possiede per "Amiga Magazine" i diritti di traduzione di **Compute!'s Amiga Resource, Compute! Publications Inc. e Transactor for the Amiga, Transactor U.K.**
Amiga Magazine è una rivista indipendente non connessa alla Commodore Business Machine Inc., né con la Commodore Italiana S.p.a.- C64 e Amiga sono marchi registrati dalla Commodore Business Machine.

Mensile associato all'USPI
Unione Stampa Periodica Italiana

Testata aderente al C.S.S.T. non soggetta a certificazione obbligatoria per la presenza pubblicitaria inferiore al 10%

Il Gruppo Editoriale Jackson pubblica anche le seguenti riviste:

Amiga Magazine Games - Pc Games - Guida Videogiochi
Supercommodore 64 e 128 - Pc Software - Fare Elettronica
Bit - Computer Grafica & Desktop Publishing - Informatica Oggi
Informatica Oggi Settimanale - NTE Compuscuola
Pc Floppy - Pc Magazine - Automazione Oggi
Trasmissioni Dati e Telecomunicazioni - Elettronica Oggi
EO News settimanale - Strumenti Musicali - Watt
Meccanica Oggi - Strumentazione e Misure Oggi

Editoriale

Insieme

Dopo mesi e mesi di ritardi, scuse eccetera, siamo orgogliosi di uscire finalmente, quasi (il ritardo è sempre in agguato) in orario, rispettando la "famosa" cadenza mensile.
Ma non vogliamo annoiarvi con i soliti discorsi "triti e ritriti" così passiamo subito a presentarvi la nostra ultima fatica, ormai questo termine è di moda !!

Nella parte TRANSACTOR vi presentiamo tre interessantissimi articoli: il linguaggio assembly parte terza, ricordiamo che le altre due sono state pubblicate su TRANSACTOR PER AMIGA, come copiare i dischi, alla scoperta del trackdisk.device, e Viewport, il nostro Phillips ci parla di un progetto di file requester di sistema.

Nello "zibaldone" di Amiga Magazine parliamo di: dispositivi di Input alternativi, Amiga TeX, un super programma per impaginare, la seconda parte di Music-X, uno speciale sulla comunicazione, e dai prossimi numeri la telematica troverà un suo spazio fisso, la recensione della guida ufficiale Commodore, un fantastico audio video digitizer e... tantissime altre notizie, senza dimenticare il dischetto allegato !!

Arrivederci a Giugno con un altro super Amiga Magazine !!

La redazione

Problemi con Hard Disk

Spett. Amiga Magazine,
sono un felice possessore di un Amiga 2000.
Recentemente ho aggiunto al mio computer un Hard Disk con autoboot.
Ebbene, tutto funziona perfettamente tranne che per un particolare: quando utilizzo la RAD: (necessaria per esigenze di lavoro) non mi è più possibile fare il boot da HD, perché il sistema si avvia direttamente da RAD:. E' possibile evitare tutto ciò?

Stefano Occhiuto

Caro Stefano, tutto è possibile! Nel tuo caso puoi aggirare l'ostacolo in due modi:

1) creare una startup-sequence appositamente per il boot da RAD: che copierai nella directory s della RAD: che avrà il compito di passare il controllo al tuo harddisk (un po' come avveniva ai tempi del kickstart 1.2, quando non era possibile fare il boot da HD)

2) puoi aggiungere, nella descrizione dei parametri della RAD:, (nel file DEVS:mountlist) la seguente linea:

```
BootPri = -127
```

In questo modo viene settata la priorità di boot al minimo possibile, cosicché il sistema si avvii dal device con il più alto grado di priorità (di solito df0:).

Il Mountlist

Spett. Redazione,
sono un giovane lettore alle prime armi che però ha iniziato subito a fare un po' di pratica con il CLI di Amiga.
Ho una domanda da porvi: che significato ha il file mountlist presente sul dischetto del Workbench ?

Massimiliano Guglielmi

Questo file viene letto automaticamente quando viene eseguito il comando Mount.
Esso permette di aggiungere nuovi device al sistema.
Per default Amiga riconosce: DF0: (drive interno), PRT: (stampante), PAR: (interfaccia parallela), SER: (interfaccia seriale), RAW: (interfaccia finestra RAW), CON: (interfaccia finestra CON) e RAM: (la Ram Disk).
Quando si desidera aggiungere un nuovo device (nuovi drive, hard disk o loro partizioni) bisogna quindi utilizzare il comando Mount che ha la seguente sintassi:

```
Mount device/A FROM/K file
```

dove device è il nome del nuovo device (ad es. DF1:, DH0:), e file è il file da cui il comando Mount prende le informazioni.
Se nessun file viene specificato, viene utilizzato il file MountList, presente nella directory DEVS:.
Vediamo ora come è strutturato il file Mountlist.
All'interno del file possiamo trovare, naturalmente, più di una descrizione di device.
Ogni descrizione di device deve iniziare con il nome del device (DF1:, DH0:, ecc.) e deve finire con il carattere '#'.
Successivamente troviamo quella che è la vera e propria descrizione del device.
Possiamo vedere nella tabella sottostante il significato dei campi nel caso si voglia aggiungere un nuovo drive.

Device	Nome del gestore del device
Unit	Numero del device (ad es. 1 per DF1:)
FileSystem	Nome del file system utilizzato (nel caso di hard disk verrà specificato FastFileSystem)
Priority	Priorità del task
Flags	Parametro per OpenDevice
Surfaces	Numero di lati del drive (per i floppy disk: 2)
BlockPerTrack	Numero blocchi per traccia
Reserved	Numero di blocchi per il Boot (di solito 2)
PreAlloc	Per il momento senza funzione
Interleave	Specifico del device (di solito 0)
LowCyl	Numero della traccia minima
HighCyl	Numero della traccia massima
Buffers	Dimensione della memoria tampone in blocchi
BufMemType	Tipo di memoria: 0 o 1 il primo tipo disponibile (CHIP o FAST) 2 o 3 solo CHIP RAM 4 o 5 solo FAST RAM
Mount	Se uguale ad 1 il device viene aggiunto al sistema immediatamente. Se uguale a -1 il device viene aggiunto al sistema al primo accesso.
Mask	Informazioni sulla zona di RAM utilizzata in funzionamento DMA.
DosType	Nel caso di FFS bisogna specificare 0x444F5301
BootPri	Indica la priorità di boot
	Nel caso si tratti di altro device (ad es. NEWCON:, AUX, ecc.):
Handler	Nome del gestore del device
Priority	Vedi sopra
StackSize	Dimensione dello stack per questo task
Mount	Vedi sopra

Il 500 e gli Hard Disk

Spett. Redazione,
vorrei porgervi i complimenti per la rivista, che, dopo il restyling, è veramente ottima. .
Passiamo al mio problema: posseggo un Amiga 500 e vorrei utilizzare un HD con autoboot.
Devo per forza comprare il Multibrain e utilizzare le costosissime schede dell'A2000, o c'è qualche altra soluzione ?
Sicuro di una risposta, porgo distinti saluti.

Matteo Sciarpa

Innanzitutto, ti ringraziamo per i complimenti.
Per quanto riguarda gli HD su Amiga 500, la Commodore ha commercializzato, già da qualche tempo, un hard disk appositamente per Amiga 500, con autoboot: si tratta dell'A590, che possiede un hard disk ST-506 da 20 Mbyte con un tempo di accesso di 80 millisecondi.
Il controller possiede, inoltre, un'interfaccia SCSI per collegare un HD di questo tipo.
Sempre nell'A590 troviamo un'espansione di memoria (autoconfigurante) di 2 Mbyte. Il tutto è naturalmente esterno e viene collegato sulla sinistra dell'A500.

Precisazioni

Per un errore di stampa volevamo precisare che il programma Kindwords 2.00 è distribuito da: C.T.O. s.r.l. - Via Piemonte 7/F- 40096 Zola Predosa (BO).

Ci scusiamo con tutti i nostri lettori.

La redazione

Collaboratori ?

La redazione invita tutti i lettori ad inviarci i propri programmi e si riserva di pubblicare e retribuire quelli che ritiene i migliori. Sono ben accetti sia programmi di giochi, sia programmi di utility.
I programmi inviati non saranno, per ragioni redazionali, restituiti.

Le opere dovranno essere inviate a:
Gruppo Editoriale Jackson
Amiga Magazine
Via Pola, 9 - 20124 Milano

Dalla redazione...

Per ovvie ragioni di spazio non ci è possibile pubblicare tutte le richieste che giungono in redazione.
Cercheremo di rispondere privatamente alle lettere non pubblicate.

International User Group

User Group del mondo unitevi !!

Questo è il motto che ha la nuova rubrica di Amiga Magazine. Se fate parte di un User Group e volete vedere pubblicato il vostro nome, inviateci i vostri dati: nome e il nome del vostro gruppo, indirizzo, telefono Bbs ecc.
Vengono pubblicati anche indirizzi di User Group stranieri in modo da aprire le frontiere fra utenti Amiga.
La redazione non si assume nessuna responsabilità sui dati inviati e pubblicati.

Il nostro indirizzo è:
Gruppo Editoriale Jackson
Amiga Magazine - Area Consumer
Rubrica "International User Group"
Via Rosellini, 12
20124 Milano

Svizzera

Amiga International User Group, Program Exchange Service, P.O. Box 18,5604 Hendschiken, CH-Swizerland

Olanda

Comiac-Software User Group, Jarrod Bernadina, Prinses Mariannelaan 320, 2275 BR, Voorburg, Olanda

Filippine

Society of Commodore Users in Metro Manila (SCUM), c/o Ruperto A.O. Navarro, 946 Murillo St., Quiapo, Manila, Philippines 1001

Australia

Commodore Computer Users Group (QLD) Inc, P.O. Box 274, Springwood Qld., 4127, Brisbane, Australia

Commodore Hornsby User Group Inc., P.O. Box 1578, Hornsby Northgate, N.S.W. 2077, Australia

Commodore User Group (ACT), P.O. Box 599, Belconnen, A.C.T., Australia 2616

Stawell Commodore User Group, P.O. Box 299, stawell, 3380, Australia

Colombia

Club Commodore Colombia, c/o Jorge Bonilla, Avenida Caracas No. 52-79, Of. 401, P.O. Box 36621, Bogota, Colombia (Sud America)

ComSoft Commodore User Group, D.F. Cardenas, Apartado Aereo 9872, Cali, Colombia (Sud America)

Messico

Club Commodore de Juarez, Calle del Manantial #1448, Ciudad Juarez, Chihuahua, Messico 64610

U.S.A.

Westside Amiga Users Group, 1814 W. 242nd Pl., Lomita, CA 90717

Amiga Users Group of Long Beach, 5155 Via Veranada, Long Beach, CA 90805

a cura di S. Leemon, M. McLean

L'eco dei nuovi prodotti

La Impulse sarebbe sul punto di commercializzare Harmony, uno strumento per il riconoscimento di tonalità del costo di 200 dollari circa. Suonando uno strumento non MIDI in un microfono, o semplicemente fischiettando, Harmony ne ricaverebbe uno spartito musicale, registrabile in formato IFF SMUS dal quale poi si potrebbero ricavare i corrispondenti eventi MIDI, editarli e stampare il tutto, facilitando in misura inedita il processo di composizione musicale. Proprio come fischiettare una canzone. Se tutto questo vi suona (scusate il bisticcio) familiare, c'è una buona ragione; quasi quattro anni fa, una società chiamata Cherry Lane Technologies pubblicizzò un prodotto con il nome di Pitch Rider, che doveva svolgere gli stessi compiti o giù di lì. La sua effettiva comparsa fu ritardata di anni, al punto che quasi nessuno riuscì a vederne un modello. Un altro prodotto Cherry Lane annunciato e poi scomparso si chiamava (avete indovinato?) Harmony, e doveva ascoltare la musica suonata nel Pitch Rider e poi ricorrere ad algoritmi di intelligenza artificiale per permettere la creazione e la modifica dello spartito. La morale? Qualche volta l'hardware e il software diventano vaporware, trasformandosi in prodotti fantasma; qualche volta, però creando le condizioni necessarie il vapore si condensa in prodotti concreti ! ▲

Un capolavoro di font

La AROCK Computer Software ha annunciato Masterpiece Professional Font Collection. I 20 dischi del package contengono 110 differenti stili di font, molti dei quali in grandi dimensioni. Il 95% dei font è infatti in corpo 100 o maggiore, anche se comunque è possibile il ridimensionamento. Tutti font sono in alta risoluzione. La raccolta comprende anche 141 pagine di documenti Deluxe Paint, con centinaia di oggetti ed esempi raffigurati. Altre opzioni includono due dischi pieni di pennelli colorati, quattro dischi di ColorFont e un manuale di 100 pagine con esempi di tutti font pronti da stampare. ▲

AROCK Computer Software
1306 E. Sunshine - Springfield, MO 65804

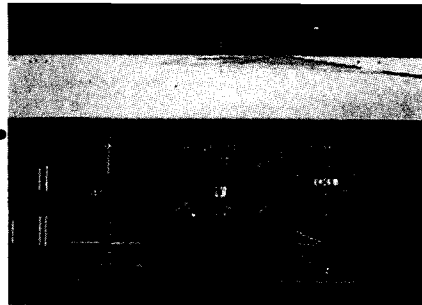
All'attacco in elicottero

In Gunship, della Microprose, prendete il controllo dell'elicottero da combattimento AH-64A Apache in dotazione all'esercito statunitense. Armato di missili Hellfire, Sidewinder e di razzi aria-aria, l'elicottero monta un cannoncino da 30 mm capace di 625 colpi al minuto.
Sono disponibili numerosi tipi di missioni, obbiettivi e livelli di gioco. Facendo del proprio meglio, si può arrivare a ottenere encomi, medaglie e promozioni.
Le missioni dei piloti si svolgono nel Sud-Est asiatico, in America Centrale, in Medio Oriente e in Europa, più uno scenario di allenamento ambientato negli USA.
La versione per Amiga è installabile su hard disk, utilizza un meccanismo di protezione dalle copie illegali e può essere usata anche con un joystick. ▲

Microprose
180 Lakefront Drive - Hunt Valley, MD 21030

Alla ricerca della Corona della Forza

In The Last Inca, avventura grafica/testuale della Free Spirit Software, il giocatore assume l'identità di un giovane guerriero Inca alla ricerca della mitica Crown of Power, la Corona della Forza, nel tentativo di precedere il mago Zulphosi, il cattivo di turno. Zulphosi, tornato dal suo esilio nelle Galapagos, intende usare la corona per distruggere il popolo Inca. Durante il viaggio, il giocatore incontrerà molte sfide, e soltanto i più dotati sopravviveranno alla marcia tra le Sacred Mountains, dimora del Great Condor, il Grande Avvoltoio.
Cammin facendo, si scopriranno tracce della civilizzazione operata dagli ormai estinti Megaterras, ed esploreranno le Caves of Doom, le Grotte del Destino, alla ricerca dello Smeraldo di Xiahuxi. ▲

Free Spirit Software
P.O. Box 128 - 58 Noble St.- Kutztown, PA 19530

Brutte copie

Dove andremo a finire? I programmi protetti hanno dato vita a una classe di software destinata a sproteggerli. Per complicare ulteriormente le cose, alcuni programmi di copia sono essi stessi protetti, per cui è nato software per la sprotezione dei programmi protetti per sproteggere... siete ancora qui? La notizia, comunque, è che pare che uno sviluppatore di copiatori, durante la sprotezione di un programma di copia di un concorrente, abbia trovato in quest'ultimo larghe porzioni di codice prese di peso da un terzo software. E questa, a suo e nostro giudizio, è veramente un'estensione un po' larga del concetto di copia... ▲

Gli insegnamenti del Guru

La Sassenrath Research ha pubblicato Guru's Guide, Meditation #1, il primo volume di una serie dedicata agli aspetti tecnici di Amiga. Il volume 1 è stato scritto da Carl Sassenrath, già autore di Amiga's Multitasking Executive Operating System. Il libro si rivolge a tutti i programmatori e intende guidarli attraverso il software di sistema di Amiga. Il libro è suddiviso in tre sezioni. Il capitolo 1, Introduction, contiene materiale introduttivo e offre un punto di partenza per la comprensione del sistema. in questo capitolo, vengono poi discussi i maggiori componenti del sistema e la storia di Amiga. Il capitolo 2, Principles, si addentra nel progetto degli interrupt e nella questione della loro importanza strategica, fornendone un'analisi approfondita. Il terzo capitolo, Programming, mette a disposizione le conoscenze di Sassenrath sulla struttura degli interrupt di Amiga. I lettori potranno ricavarne esempi e tecniche d'uso estremamente utili, e una guida alla creazione di applicazioni "well-behaved", che rispettino le specifiche software per realizzare un programma. ▲

Sassenrath Research
P.O. Box 1510
Ukiah, CA 95482

Rete!

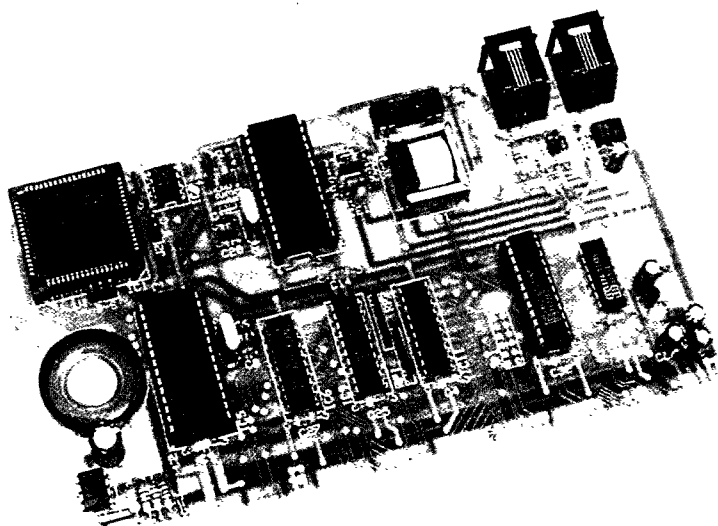
Un'area in cui Amiga deve decisamente fare progressi è quella delle LAN, le Local Area Network o reti locali. Una rete migliora l'utilizzo di un computer in molti modi. Al livello più basso, permette a molte macchine di condividere l'uso di periferiche costose, come stampanti laser o megadischi rigidi. Configurazioni più sofisticate permettono trasferimenti di file in trasparenza da un Amiga all'altro e la posta elettronica. Le applicazioni più avanzate permettono di collegare Amiga ad altri microcomputer, mini e mainframe.Sorprendentemente, Amiga poteva effettuare quest'ultimo tipo di connessioni quasi dalla sua nascita, grazie all'adattatore Ethernet Ameristar. Sebbene costoso (900 dollari), è ideale per università e per chi abbia esigenze di collegamento a workstation Sun.
Pare che Commodore stia cominciando a recepire il messaggio dell'utenza a proposito delle LAN, e sembra che esistano alcuni progetti in fase di concepimento. I pettegolezzi riguardano Ameristar (che fornirebbe con la sua scheda Arcnet un collegamento economico Amiga-Amiga) e TOPS, una divisione della Sun famosa per l'omonima rete di collegamento tra sistemi IBM, Macintosh e Unix. ▲

Dopo Tuono Blu...

Il videogioco da bar Thunder Blade della Sega è stato convertito dalla Mindscape in una versione per Amiga. Il gioco offre due differenti viste, da sopra e dalla parte posteriore dell'elicottero. La vista da sopra permette di esaminare il paesaggio in 3D sottostante, mentre quella da dietro consente di volare verso la destinazione in vista. Il pilota dovrà vincere i duelli aerei ambientati in paesaggi a base digrattacieli, montagne, deserti e oceani. Oltre a Thunder Blade, la Mindscape produce altre riedizioni di videogiochi Sega, come Out Run. Nel futuro sono previste versioni Amiga di Shinobi, Action Fighter e After Burner. ▲

Affari interni

Supra ha presentato il modem interno SupraModem 2400zi per i computer della serie Amiga 2000.
Il modem, compatibile Hayes, è una "half card" che supporta la comunicazione asincrona a 300, 1200 e 2400 baud.
Il modem è configurabile con sequenze di comandi, che possono essere immagazzinate nella memoria non volatile del modem e caricate al momento dell'accensione.
Il SupraModem trova posto in qualunque slot di Amiga, ed è compatibile con tutti pacchetti di telecomunicazione più diffusi.
Le opzioni a disposizione comprendono autoanswer e autodial, due jack per la connessione alle linee telefoniche, un altoparlante con volume regolabile e software che permette a più modem il collegamento con lo stesso computer.
Il pacchetto è completo di modem, cavo telefonico, prontuario dei comandi, manuale e un certo tempo di collegamento prepagato per le maggiori banche dati americane. Unico requisito, il possesso di un qualunque programma di telecomunicazione. ▲

Supra
1133 Commercial Way
Albany, OR 97321

Joe Blade entra in azione

Sei leader politici mondiali sono stati rapiti, e un solo uomo è in grado di salvarli: Joe Blade. In questo gioco della DigiTek Software, il giocatore cerca di fare entrare Joe nella fortezza del malvagio Crax Bloodfinger. Una volta penetrato all'interno, bisogna frugare nelle 127 stanze del palazzo per trovare le chiavi necessarie, liberare gli ostaggi, disinnescare alcune bombe e uscire. ▲

DigiTek Software
8910 N. Dale Mabry Executive Center
Suite #37
Tampa, FL 33614

Il sistema segreto

Si vocifera da tempo di nuove macchine da gioco prodotte congiuntamente da Commodore ed Epyx, basate su Amiga.
Epyx ha ora presentato una nuova macchina da gioco, a porte chiuse. Varie persone hanno potuto vederla, ma al momento il segreto è ancora ben custodito. Vincolati alla riservatezza da un accordo di 15 pagine, molti dei... pochi fortunati hanno commentato "Non è ciò che ci si aspettava".
Altri hanno dichiarato "Quello che è stato detto finora è probabilmente vero", il che tende tende a confermare l'ipotesi che non sia effettivamente quello che ci si aspettava.
Sembra dunque svanire l'idea dell'utilizzo della tecnologia di Amiga in questa direzione, anche se la filosofia di progettazione potrebbe essere simile.
Un'altra ipotesi che probabilmente verrà smentita è che si tratti di un sistema video interattivo basato sulle tecnologie CDI, VDI e DVI.
La voce che ci sembra più realistica è quella di un qualche genere di macchina da gioco tascabile.
A differenza di quelle viste finora, però, questa non avrebbe un banale schermo LCD 30 x 30, ma un vero video a colori ad alta risoluzione.
Potrebbe essere insomma una specie di Watchman (il televisore da passeggio della Sony) per videogame. una scommessa contro la rivale Nintendo, la cui posta in gioco potrebbe essere un mercato di dimensioni favolose. Pensate a quanti giochi da tasca avete già visto in mano a bambini... ▲

Articolo tradotto dal n.3 Fall 1989 di Compute!'s Amiga Resource.

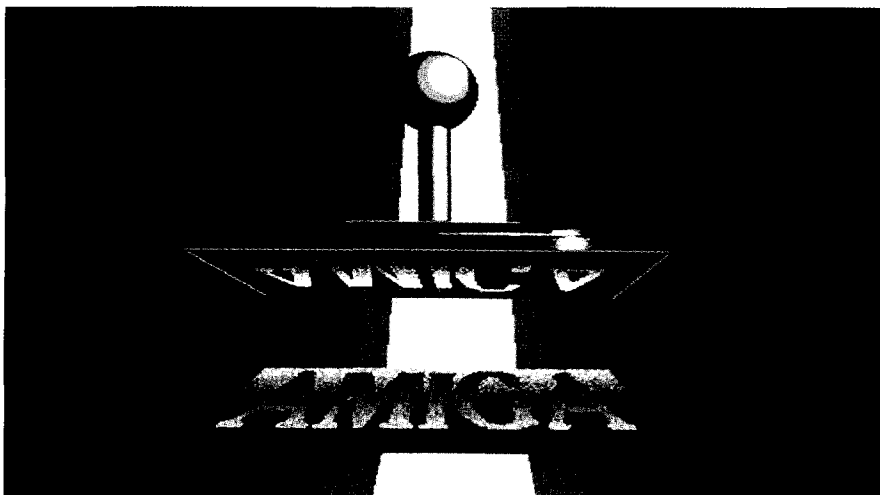
Personalizzare Amiga

Rhett Anderson & Randy Thompson

L' interfaccia utente di Amiga è veramente unica. Potete usare il computer operando in grafica, nel Workbench, o in modo testo da CLI, o da entrambe insieme. Qualunque modo scegliate, potete sempre personalizzare l'ambiente in cui vi trovate. Questo articolo vi accompagna, passo dopo passo, alla scoperta dei sistemi per realizzare esattamente il "vostro" ambiente di lavoro su Amiga.

L'interfaccia utente di Amiga è in costante evoluzione. I primi Amiga 1000 incorporavano la versione 1.0 di Kickstart e Workbench. A essa succedette la 1.1, anch'essa con gli stessi limiti: bug perfino noiosi e una legnosità senza pari. La versione 1.2 era invece decisamente migliore, essendo la prima sufficientemente stabile da essere inserita in ROM. Molte cose sono cambiate dalla 1.0 alla 1.2, ma fondamentalmente l'interfaccia utente è rimasta la stessa.
Il livello più profondo dell'interfaccia utente è chiamato Intuition. Esso controlla gli oggetti più comunemente utilizzati dagli utenti: schermo, finestre (l'uno e le altre con i relativi gadget), menu e sottomenu. Sopra Intuition poggiano due ambienti di lavoro: Workbench e CLI. Il primo è la vera e propria interfaccia utente grafica, Macintosh-like; la seconda è l'interfaccia testo, MS-DOS-like, ed entrambe possono essere usate, anche nello stesso momento, grazie alle capacità di multitasking di Amiga. Volendo, è possibile attivare più di una CLI (non più di un Workbench, comunque).
Attualmente il software di sistema è arrivato alla versione 1.3, che permette l'avviamento del computer da dispositivi diversi da un floppy, e la 1.4 (che potrebbe apparire il prossimo autunno) dovrebbe apportare decisi miglioramenti, quanto mai necessari, al look del Workbench.
Non solo Commodore, comunque, sta lavorando alla crescita dell'interfaccia utente; sulla stessa strada sono impegnati numerosi programmatori di altre case o indipendenti.

Un Workbench più bello

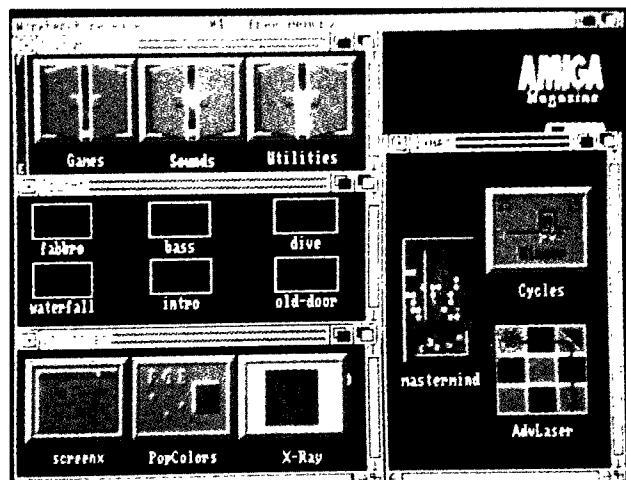
La maggior parte delle modifiche al Workbench sono di carattere estetico. Usando Preferences, è possibile cambiare i colori dello schermo, il numero di colonne (60 o 80) e l'aspetto del puntatore. L'utility FastFont consente di sostituire il Topaz, il font di sistema del computer, con un altro font non proporzionale di propria scelta (FastFont) ed è anche disponibile, come parte di TexEd, il pacchetto della Microsmith. Basta un doppio click sull'icona di ClockPtr, nella directory Utilities di Workbench 1.3, e il puntatore diviene un orologio digitale che si aggiorna costantemente, per tenere informato l'utente. E questi sono solo alcuni degli elementi modificabili.
Tutte le icone possono essere alterate a piacimento, con i programmi IconEd e IconMerge situati nella directory System del Workbench 1.2 o nei Tools della versione 1.3. IconEd è un banale programma di disegno, per l'editing delle icone; IconMerge fonde insieme due icone separate, creandone una sola, animata, con una immagine visualizzata in caso di selezione e l'altra mostrata normalmente.
L'uso di IconEd può essere comunque frustrante. Il miglior modo di personalizzare le icone è procurarsi un programma di disegno di qualità professionale, come Express Paint 3.0, della Brown-Wagh. Quest'ultimo ha la possibilità di caricare e salvare e icone con la stessa facilità con la quale vengono caricate o salvate le immagini. Se usate un programma come DeluxePaint,

avrete bisogno di una utility di conversione, per trasformare i file IFF in icone, come Brush2Icon.
Disegnando le vostre icone, vi capiterà di sentirvi limitati dai soli quattro colori permessi dal Workbench. Bene, nel numero 10 di Amiga Magazine si trova un programma, Colorbench, che elimina questa limitazione portando i colori a un massimo di otto.
E se siete stanchi dello sfondo monocolore e monotono del Workbench, ci sono molti programmi shareware e di pubblico dominio in grado di aiutarvi. Rainbench, per esempio, crea sullo sfondo una cascata iridescente di colori.
Con DropCloth, potete piazzare sullo sfondo retini e perfino immagini IFF. Per altri effetti, Wavebench crea ondate marine, mentre DropShadow aggiunge l'ombra a tutte le finestre.
Il software di pubblico dominio costituisce un'ottima fonte di programmi ingegnosi e coloratissimi per personalizzare il look di Amiga.

CLI in cambiamento

Nonostante le attrazioni grafiche del Workbench, la CLI è usata, almeno parzialmente, da moltissimi possessori di Amiga. Per la sua stessa natura di interfaccia a comandi, la CLI è un ambiente in evoluzione. Si può usare il comando Rename per cambiare il nome dei comandi stessi, o cancellarli, aggiungerne o modificarli, oltre che sperimentare ogni sorta di trucco programmatorio.
Sebbene non si possa fare molto

Progettare icone

Troy Tucker

Dopo molte ore di lavoro, finalmente il programma è completo e non vedete l'ora di mostrarlo ai vostri amici. L'unica cosa che resta da fare è copiare il programma su un altro disco e rubare un'icona del Workbench, questa volta, magari, quella dell'orologio. Una volta cambiato il nome, tutto è pronto.
Avete già fatto qualcosa del genere? Perché usare un'icona del Workbench invece che disegnarne una propria, volendo anche animata? Una volta imparati alcuni trucchetti, si rivela un compito facile. Prima di tutto, servono un paio di strumenti indispensabili, di cui il primo è un buon programma per disegnare, che potrebbe essere DeluxePaint III, ma anche qualunque altro di vostra preferenza, purché faccia uso del formato IFF Poi è necessario un programma di conversione IFF-icona. Se non ne avete uno, probabilmente potete procurarvelo presso qualche BBS oppure utilizzando IFF to Icon, routine presente su molti dischi di Public Domain. Se desiderate animare le icone, vi serve anche IconMerge, contenuto nel System di Workbench 1.2 e nei Tool della versione 1 3.
La prima cosa da fare per realizzare un'icona è creare l'immagine che meglio ricorda il vostro programma, o che ha il significato che desiderate. Non ci sono limiti all'immaginazione, ma non esagerate nelle dimensioni, per non dilatare il tempo di caricamento da disco. E cambiate la vostra palette di colori, per adattarla allo schermo del Workbench. Siccome le immagini saranno visualizzate sui 640 x 200 pixel del Workbench, disegnate l'immagine usando lo stesso modo grafico e solamente i primi quattro colori della palette, pena risultati veramente poco soddisfacenti e icone dall'aspetto decisamente insolito.
Dopo avere disegnato l'immagine, salvatela su disco come brush. Ricordate che il testo dell'icona verrà piazzato direttamente sotto l'immagine, per cui lasciate un paio di righe di pixel vuote alla base del disegno; il risultato finale sarà più elegante.
Se state creando un'icona animata, ricordate che i brush delle due immagini necessarie (quella "ufficiale" e quella attivata dal click del mouse) devono avere la stessa dimensione. Non è necessario invece che i due disegni siano della stessa grandezza. Pensate a un'icona che mostri, per esempio, un libro chiuso che, cliccando con il mouse, si apre. E' ovvio che l'immagine del libro chiuso sarà circa metà dell'altra e che, nel disegnare il libro stesso, ci preoccuperemo di lasciare nel brush lo spazio sufficiente per ospitare, nella seconda immagine, la parte di libro (aperto) prima non esistente. Suggeriamo di cominciare a disegnare all'interno di due rettangoli o quadrati di uguale grandezza, per essere sicuri che le due immagini abbiano non solo le stesse dimensioni, ma anche la stessa posizione all'interno dell'icona. Ora si può procedere alla conversione dei disegni IFF in icone. Prima di lanciare il programma di conversione, è necessario duplicare un paio di file .info. Infatti, la maggior parte delle routine di conversione scrivono direttamente nel file dell'icona, cambiando soltanto i dati dell'immagine Così, se l'icona deve indicare un drawer, dovete duplicare il file .info di un drawer, se contraddistingue un programma bisogna fare una copia del file .info di un programma, e via duplicando. Per essere sicuri di avere il giusto tipo di icona, selezionatela con il click del mouse e chiedete INFO dal menu Workbench. Il campo TYPE di un programma contiene la parola Tools, quello dei dati Project, quello di un cassetto la scritta Drawer e quello di un disco la parola Disk. Dopo avere attivato il programma di conversione, l'icona dovrebbe essere pronta. Se l'icona è animata, andrà divisa nelle sue due parti, da riunire con IconMerge. Occorre a questo proposito scegliere il menu M (da Merge) e inserire il nomefile della prima icona, quello della seconda e il nome finale della nuova icona animata. Ricordate di NON includere l'estensione .info nei nomifile. Dopo avere riunito le due icone, IconMerge si interromperà lasciandovi nel Workbench. Per vedere la nuova icona, bisogna chiudere e poi riaprire il disco. Maggiori particolari su IconMerge sono rintracciabili nel manuale Enhancer Software 1.2 o 1.3.

per modificare l'aspetto grafico di una finestra CLI, se ne possono cambiare nome e dimensioni di schermo editando il file s:CLI-Startup. Per esempio, si può digitare WINDOW = CON: 0/10/640/100/ MyCLI, per aprire una finestra chiamata MyCLI nella parte alta dello schermo.
Nella versione 1.3 al posto della CLI compare un suo sostituto, lo SHELL. SHELL è abbastanza simile a CLI, ma offre una grande quantità di nuove opzioni. Con lo SHELL è possibile editare i comandi digitati, e perfino cercare comandi che comincino con un'altra sequenza di caratteri. In aggiunta, SHELL permette di creare comandi residenti, che non hanno bisogno di essere caricati ogni volta da disco. Si possono cambiare nome e dimensioni di SHELL editando il file s:SHELL-Startup. Aggiungendo l'istruzione WINDOW = NewCON: 0/10/640/100/ MyShell si creerà una finestra di 640 x 91 pixel, chiamata MyShell. SHELL permette anche di personalizzare il proprio ambiente con il comando Alias, che può essere utilizzato per ottenere i comandi convenzionali in forma abbreviata. Per esempio, il comando ALIAS D1 "DIR DF1:" permette di digitare solo D1 e avere la directory del drive DF1:. Sebbene si possano digitare gli alias direttamente dalla command line, ha più senso inserirli nel file SHELL-Startup.
Come già detto, SHELL dà anche la possibilità di disporre di comandi residenti, operanti senza bisogno di essere caricati da disco ogni volta. Come si può immaginare, questo velocizza notevolmente l'utilizzo di Amiga.
Normalmente, per avviare CLI o SHELL bisogna cliccare due volte sull'apposita icona. Tuttavia, cancellando l'istruzione EndCLI nel file di startup, potete tenere la finestra già aperta sullo schermo. Oppure ricorrere a PopCLI, un programma che richiama CLI attraverso la semplice pressione di un tasto. PopCLI è costruito per richiamare CLI, ma gli utenti SHELL possono rimediare facilmente cambiando il comando di installazione in POPCLI 180 SYS:C/NEWSHELL> NIL:NEWCON:0/12/640/188/ MyShell. CLI e SHELL utilizzano come prompt un numero progressivo. Potete cambiare il prompt utilizzando l'apposito comando. Per esempio, PROMPT "Cosa devo fare?" ingentilisce un pochino il rapporto con il computer. I parametri %N e %S visualizzano infine il numero progressivo e la directory attualmente in uso. Provate a digitare PROMPT "CLI %N, PATH %S>".

Browser, alternativa al Workbench

Tra i vari ambienti alternativi al Workbench, quello più convincente rimane Browser. Molto simile al Workbench, Browser ne differisce per il fatto di lavorare con i nomi dei file anziché con le icone. Questo permette di lavorare indifferentemente con file provvisti o privi del loro file .info. In più, con Browser, le applicazioni di maggiore utilizzo possono essere inserite in un menu a discesa direttamente accessibile, per evitare di dover rintracciare ogni volta i programmi in mezzo alle directory più disparate. ▲

ARP, e il DOS rinasce a nuova vita

Rhett Anderson

Nonostante le varie revisioni, AmigaDOS non è sicuramente alla sua versione definitiva e presenta ancora qualche problema di funzionamento. Un team di programmatori Amiga guidato da Charlie Heath ha così pensato di lavorare sul nucleo dei comandi CLI di AmigaDOS, per riscriverlo. Il risultato è ARP: l'AmigaDOS Replacement Project, o progetto per la sostituzione di AmigaDOS. Per rendere quanto più possibile affidabili e di ridotte dimensioni i comandi, il team ARP ha realizzato una libreria (ovviamente chiamata arp.library) di comandi progettati per estendere le possibilità della tradizionale libreria dos.library e accessibili da qualsiasi programma, dietro disponibilità del file arp.library all'interno della directory libs del disco Workbench.
Una delle opzioni più utili di ARP è il cosiddetto file requester, una finestra che consente di selezionare file da cancellare, copiare, caricare, registrare e via discorrendo (e che Amiga avrebbe dovuto avere fin dalla nascita). Il team ARP spera così di ovviare alla situazione attuale in cui, vista la mancanza di un file requester standard, ogni applicazione si comporta a sua totale discrezione, complicando non poco la vita degli utenti.
Funzioni avanzate a parte, anche per l'utente medio ARP si dimostra utilissimo, fornendo comandi CLI più piccoli, veloci e versatili, per esempio con un migliore trattamento delle wildcard (i comandi ARP si accontentano della wildcard *, dove AmigaDOS richiede invece #?). Nella tabella sottostante troviamo alcuni confronti tra le dimensioni in byte di alcuni comandi di AmigaDOS 1.3 e dei corrispondenti ARP:

Comando	AmigaDOS	ARP
Addbuffers	876	492
Assign	3008	1264
Avail	1964	692
CD	1756	580
Copy	9848	2772
Delete	6124	932
Dir	8772	2188

La tabella non ha bisogno di essere commentata, ed evidenzia il lavoro compiuto dal team ARP in modo più che convincente. Commodore ha promesso nel frattempo una versione 1.4 di AmigaDOS più compatta e potente, presumibilmente, tra le altre migliorie, con un file requester incorporato. In attesa che la promessa diventi realtà, ARP è la scelta più logica per chi voglia migliorare le prestazioni del proprio AmigaDOS.

Articolo tradotto dal n.3 Fall 1989 di Compute!'s Amiga Resource.

Un Amiga in ogni ufficio

Rhett Anderson Vs. Randy Thompson

NO GRAZIE! E' molto tempo che Commodore cerca di far passare Amiga per un computer per tutti gli usi, senza vendere un solo computer in più. Amiga sa vendersi, invece, benissimo da solo.
Pensate ai possessori di Amiga che conoscete. Sono stati convinti a comprarlo dai maghi del marketing Commodore? No. Fortunatamente per Commodore, chi ha comprato Amiga era abbastanza intelligente per accorgersi delle possibilità della macchina.
Ora, se dobbiamo credere a ciò che si sente, Commodore intenderebbe puntare la sua attenzione al mercato professionale. Ottima mossa, ragazzi. Perché non vi dimenticate degli uffici e guardate un po' ai cloni di PC, che stanno invadendo le case di tutto il mondo civilizzato a un ritmo soffocante? E' sciocco mandare soldati all'estero quando c'è una guerra da combattere in casa. Occupatevi di ciò che conoscete meglio.
Amiga non è fatto per essere una macchina professionale. Non che l'hardware sia carente, è il software che non tiene il passo. Pensate che i maggiori produttori di software siano disposti a sacrificare risorse (già ripartite tra PC e Macintosh) per sviluppare prodotti per Amiga? Non in questa vita.
E non è solo la sezione marketing della Commodore a sbagliare direzione. Commodore sta spendendo in ricerca e sviluppo soldi preziosissimi, per costruire computer che solo una minoranza di utenti dal portafoglio "sviluppato" può permettersi. Facciamo entrare Amiga in più case. Siamo tutti interessati ai nuovi modelli di fascia alta, come il 2500, ma perché non pensare anche a computer migliorati nelle prestazioni e disponibili a prezzi competitivi? Commodore, con chi stai cercando di competere in realtà? Con IBM e relativi cloni? Buona fortuna. Specie adesso che un AT si può acquistare a prezzi praticamente irrisori, e dispone del software professionale realmente adatto per quel tipo di utenza. Con Apple? Scordatelo. Apple ha grandi risorse finanziarie, e aggiorna i suoi computer quasi con lo stesso ritmo di evoluzione del software, ogni sei mesi. Credi davvero di riuscire a fare lo stesso? Perfino se Commodore potesse davvero introdurre Amiga in ufficio, la situazione non sarebbe buona. Guardate gli utenti Macintosh. Non molto tempo fa erano ancora ribelli, promotori del computer "for the rest of us".
Oggi sono tutti in doppiopetto, e sulle riviste dedicate ci si chiede quale sia il miglior database per Macintosh. Per Amiga, solo l'estinzione sarebbe una fine peggiore. Il punto di vista di Mr. Thompson è "Amiga è un computer professionale". Mi spiace, ma penso che non sarà mai così. Vedete, è da molto tempo che Commodore non fabbrica più macchine per scrivere, calcolatrici tascabili e PET.
Ora costruisce home computer. E Amiga, come home, è perfino meglio del Commodore 64.

NE PRENDO DUE! Amiga è un computer professionale. Amiga è un computer professionale. Ripetetevelo fino a quando non ne siete convinti. Dopo tutto, Commodore continua a significare Commodore Business Machines, una società che ha costruito il proprio successo iniziale su macchine per scrivere, calcolatrici tascabili e una macchina come il PET, il primo ingresso di Commodore nel mercato dei personal computer.
Adesso abbiamo Amiga, un computer con una grande crisi: è un vero calcolatore o una macchina per giocare? Una recente pubblicità Commodore parlava di Amiga come "il computer con una mente da musicista". Cosa?!? Non so il vostro, ma il mio Amiga non ha nulla in comune con Jimi Hendrix o i Tears For Fears o chi volete voi.
Invece di stabilire analogie che

provocano solo una grande confusione, Commodore dovrebbe ribadire fatti, uno dei quali è che Amiga è un potente computer professionale.
Se Amiga verrà accettato negli uffici, disporremo di più supporto per il suo software, e Commodore ricaverà maggiori entrate che gli permetteranno di produrre macchine migliori a prezzo inferiore (un Amiga 2000 in un ufficio offre lo stesso rapporto prezzo-prestazioni di un 500 in casa).
Commodore dovrebbe richiamare l'attenzione sulle doti che rendono Amiga un computer unico: grafica, suono, multitasking e così via , senza collocarlo in una nicchia di mercato presentandolo come uno strumento per la grafica o una macchina per giocare. Bisogna far capire al grande pubblico tutto quello che questo computer può fare, non tenerlo nascosto. Guardiamo le specifiche: Motorola 68000 a 16/32 bit; fino a 9 Mbyte di RAM; tastiera a 94 tasti con tastierino numerico separato; floppy disk drive da 880 Kbyte; sistema operativo multitasking; porta seriale RS-232 e porta parallela Centronics... sembra davvero una macchina per giocare?
Alla Apple ormai sono tutti yuppie, e visto che Amiga costa poco pensano che non possa essere un "vero" computer. I manager, abituati ai PC IBM, si preoccupano del suo aspetto (perché Amiga non assomiglia a una cassetta per gli attrezzi di metallo grigio). A dispetto di questi "difetti", Amiga è un ottimo computer per il lavoro come per il gioco. Mr. Anderson, come al solito, considera la faccenda dal lato paranoico della situazione: i cloni IBM si infiltrano in tutte le camerette di tutte le case; Commodore si sta ritirando (troppo odore di PC), siamo tutti in pericolo di diventare esseri amorfi in doppiopetto grigio, meglio l'estinzione che la cravatta seria. Calma; i computer professionali non sono l'invenzione di una banda di sovversivi. Non è stata cominciata nessuna lotta armata, e nessuno sta compilando una lista degli home computer da distruggere o internare... ▲

Speciale Comunicazione

Emmanuele "Exedre" Somma

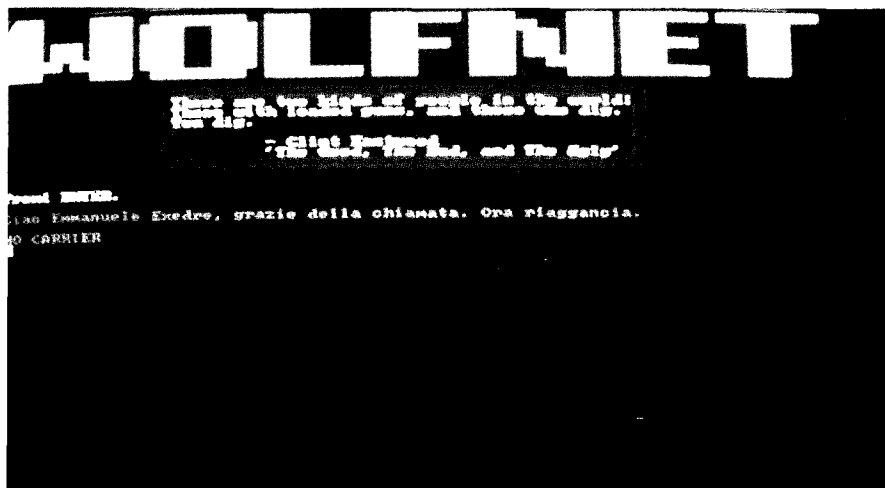
Questo articolo nasce dalla constatazione che se da un lato è abbondante la letteratura specialistica relativa al mondo della telematica, alla teoria delle comunicazioni tramite calcolatori, ai modem, ai programmi di telecomunicazione, alle banche e basi di dati, dall'altro manca una guida, semplice fino a sembrare banale, per non addetti ai lavori e per tutti coloro che proprio per mancanza di questa informazione basilare sono costretti a tenersi lontani da questo mondo che è, non solo utile e "moderno" ma, soprattutto, divertente.
Allora partendo da questa constatazione, e rilevando la sempre maggiore importanza delle telecomunicazioni in ambito domestico, l'intento di questo articolo è, soprattutto, quello di stimolare l'interesse del lettore alle enormi potenzialità della telematica e, nello stesso tempo, invogliarlo a fruirne direttamente senza relegarle esclusivamente al mondo strettamente specialistico degli addetti ai lavori.

Perché comunicare

La voglia di comunicare nasce dalla necessità, profondamente umana, di conoscenza, di confronto, e di scambio di idee.
L'esigenza di comunicare nasce dalla necessità di passare in tempi brevissimi notizie utili, di discutere in tempo reale metodi e mezzi per compiere particolari operazioni, per scongiurare comuni pericoli.
La necessità di comunicare nasce dall'esigenza di tenersi sempre informati, ma con giudizio, di interagire con le fonti di informazione e non solo di assorbire passivamente notizie, troppo spesso poco sincere, ma che prepotentemente si arrogano il diritto di essere obiettive.
La voglia di comunicare nasce dall'esigenza di sentirsi cittadini di tutto il mondo, di ridurre le distanze.
L'esigenza di comunicare nasce dalla voglia di giocare. Giocare in comunicazione con computer nel mondo che simulano interi castelli fatati e cantine segrete e Draghi e Stregoni, e scoprire che il nano che l'altroieri ti voleva decapitare con la sua ascia bipenne col manico intarsiato di diamanti e che oggi hai di soppiatto spinto in una pozza di chissà quale acido, era un ragazzo del Massachusset di Nome Irwin che ti ha scritto una enorme lettera con tutti i suoi dati, i suoi hobby, e tutto quello che a lui piace.
La necessità di comunicare nasce dalla voglia di divertirsi in compagnia, di conversare amabilmente in ogni lingua con persone al di là degli oceani, dalla voglia di conoscere persone nuove, di diverse culture, di diverse civiltà.
Parlare, scrivere, mandare brevi note e intrattenere lunghe corrispondenze eliminando i tempi morti postali spedire messaggi nuovi ogni giorno e ogni giorno riceverne di nuovi trovando la propria cassetta della posta piena zeppa di lettere digitali. Vivere, insomma, anche con il calcolatore e il modem potrebbe risultare una esperienza nuova ed eccitante, ma potrebbe anche entrare a far parte stabilmente della vita.

Come comunicare: l'Hardware.

Che modem comprare? Questo è un dilemma profondo, quella dell'hardware è spesso una giungla e i modem non fanno certo eccezione: mille diverse scatolette tutte etichettate da sigle poco distinguibili divengono collettivamente vuote e inutili, inutilizzabili e lontane. Scopriamo come ci si salva...
Solo poche cose importanti:

La velocità di trasmissione è in

pratica l'unica cosa da scegliere, e da scegliere sia in base alla disponibilità finanziaria per comprare l'apparecchiatura vera e propria, sia quella per pagare le bollette telefoniche. Un modem più veloce costa di più, ma fa risparmiare sulle bollette telefoniche (non è come una Ferrari che costa di più e consuma di più). Un modem a 300 baud trasferisce i dati 4 volte più lentamente di uno a 1200 e 8 volte di uno a 2400. Una volta scelta la velocità bisogna stare attenti che sulle caratteristiche del modem, che comprerete, compaiano esattamente i seguenti codici:

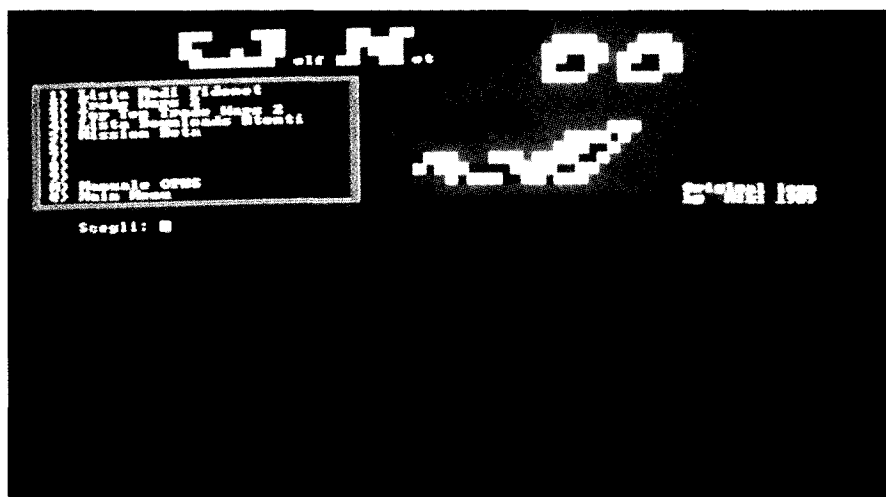
CCITT

V21	per i modem a 300 baud
V22	per i modem a 1200 baud
V22bis	per i modem a 2400 baud
V23	per il VideoTel Sip

La presenza di queste sigle, che ci assicurano la compatibilità dei modem alle norme internazionali che regolano le trasmissioni telefoniche, è quindi essenziale.

Lo Standard Hayes Esteso è indispensabile! Un modem Hayes è totalmente controllabile dal computer così che non bisogna manipolare, deviare, controllare e così via. Con un modem Hayes devi solo connettere il modem e accendere: il resto lo farà il tuo software, e a te non resterà che divertirti. Un modem Hayes deve avere anche delle spie luminose (led) e un altoparlante interno che, oltre ad essere veramente utili per capire se c'è qualcosa che non va, sono anche divertenti: io non rinuncerei mai a lasciarmi trasportare nel mondo dei sogni dalle lucine del Transmit e del Receive Data durante un trasferimento.
Lo Standard Hayes è il più diffuso: la maggior parte degli utenti di servizi telematici lo usano e ne sono felici perché essendo uno standard, molte volte si possono trovare già pronte le soluzioni ai problemi.

L'interfaccia

I modem si comandano al limite con soli sette fili, ma per una questione di semplicità richiedete una interfaccia standard RS-232C e che il connettore sia un DB-25S (lo stesso tipo di connettore a vaschetta a 25 poli che è presente dietro Amiga). A meno che non si sia capaci di costruirselo da solo, bisogna fornirsi anche del cavo di connessione con le due interfacce (non bisogna collegare direttamente il modem al computer, è scomodo e pericoloso).

Optional

Gli unici veramente utili sono i circuiti di correzione degli errori e di compressione automatica dei dati chiamati rispettivamente MNP4 e MNP5, ma questi fanno lievitare notevolmente il prezzo del modem e se l'uso del modem non è veramente intenso non giustificano una spesa così elevata. Comunque l'MNP5 riesce, sotto alcune condizioni, a raddoppiare la velocità del modem e comunque c'è sempre almeno una diminuzione del tempo di trasmissione pari al 40%.

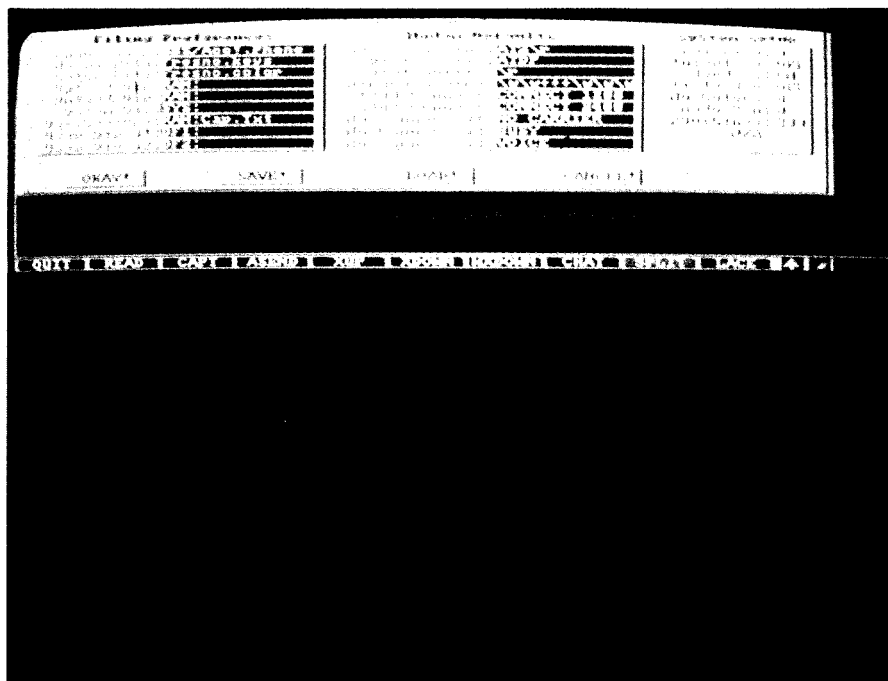
Allora in ultima analisi, comprando il modem chiedete la vostra velocità desiderata, controllate che il codice CCITT corrispondente faccia parte delle caratteristiche, verificate che il modem sia controllabile tramite i comandi Hayes Estesi (Extended Hayes "AT" command set), cercate le spie, l'altoparlante, controllate l'interfaccia, insomma non comprate le cose a scatola chiusa!

Software di comunicazione

Adesso la parte Hardware del nostro piccolo sistema telematico è a posto e funzionante. Abbiamo trovato il programma di comunicazione che fa per noi, e se non sapete dove cercarlo provate nel dischetto allegato, dobbiamo solo cominciare. Ma non è tutto così semplice: non basta arrivare in un paese straniero per poterci vivere, bisogna imparare la lingua, ed è quello che proveremo a fare adesso. Non chiedetemi perché, ma questo mondo è veramente complicato: pare che i costruttori e gli utilizzatori di apparecchi telematici non abbiano affatto voglia di mettersi d'accordo per parlare tutti una stessa lingua e, quindi, sorgono enormi difficoltà.
Non tanto enormi in realtà, ma noiose sì; ci costringono ad imparare tante altre cose che, a dire la verità, non sono affatto necessarie e noi che siamo abituati ad Amiga dove tutto è così semplice e bello, non possiamo che dolercene. Comunque, ancora per un po' di tempo sarà necessario avere almeno una idea su cosa significhino alcuni termini un po' specialistici.
Prima di poter comunicare dovremo definire i parametri della nostra comunicazione. E qui le cose diventano un po' difficili. Poniamo il caso di voler fare un collegamento con

una Banca Dati. Ammettiamo di avere un modem a 1200 baud che supporti gli standard CCITT V21, V22, V23. E' indispensabile sapere: a quale velocità massima la Banca Dati può rispondere cioè quali standard supporta; quanti bit di dati, che tipo di parità e quanti bit di stop richiede. I numeri di telefono e i parametri di tutti i BBS della rete Fido sono presenti nella tabella a pagina 20.
Prendiamo, per esempio, il BBS WolfNet di Pisa nodo della rete Fido numero 2:332/23: la velocità massima a cui arriva è i 2400 baud e gli standard supportati sono V21, V22, V22bis, V23; i settaggi del frame di dati (così si chiama il pacchetto che viaggia sulla linea telefonica composto dal bit di start, dai bit di dati, dai bit di parità, e, infine, dai bit di stop) sono 8 bit dati, Nessuna parità, 1 bit stop, o come si indica correntemente 8N1. Anche se non sappiamo esattamente cosa significhino questi codici per ora accontentiamoci di utilizzarli.
Allora dobbiamo configurare il nostro software di comunicazione con gli stessi parametri del BBS. Caricato il programma dovremo dirgli, nel menu di configurazione del modem, che il nostro può arrivare fino a 1200 baud (Baud Rate: 1200) poi dovremo settare 8 Bit di Dati (Data Bits: 8), Nessuna Parità (Parity: NONE), e 1 Bit di Stop (Stop Bits: 1). Solo a questo punto potremo chiamare il BBS.
I programmi di comunicazione si assomigliano un po' tutti poiché in fondo il loro compito è abbastanza limitato. Comunque sia alcuni programmi risultano molto validi per la loro flessibilità e per alcune opzioni particolari. Mentre non è necessario che un modem abbia tanti optional, proprio perché questi possono venire egregiamente simulati dai software, è, invece, indispensabile che il programma di telecomunicazione permetta di fare alcune cose in modo semplice e diretto. Sono utilissime, quindi, opzioni come la filtratura dei caratteri di controllo (spesso denominata emulazione di terminale), l'agenda telefonica completa oltre che dei numeri anche di tutta la configurazione, la definizione dei tasti di funzione, la "cattura" dei testi su di un file, un buon numero di protocolli di trasmissione. Ma, di tutte, forse la capacità più utile è quella di eseguire script, veri e propri programmi all'interno del software di comunicazione per rendere più semplici i collegamenti e per poter eseguire alcune operazioni in modo del tutto automatico.

Che cosa è un BBS?

La sigla BBS sta per Bullettin Board System cioè più o meno quel sistema di bacheche che nelle Università servono per gli annunci, per la compravendita di oggetti usati, per le offerte di lavoro, per i messaggi sia personali sia diretti a tutti coloro che si trovino a leggerli.
La novità sta nel fatto che all'antica bacheca di sughero è stato sostituito il computer, accessibile a tutti tramite la normale linea telefonica.
I BBS qualitativamente migliori si riuniscono in una rete telematica chiamata FidoNet. FidoNet è una struttura stabile e molto funzionale, tenendo conto che è assolutamente amatoriale, cioè che tutti gli operatori dei sistemi di BBS della FidoNet non sono pagati da nessuno per tenerli attivi ma lo fanno esclusivamente per passione. Qualche cifra: la rete Fido conta nel mondo migliaia di nodi (con nodo si intende un BBS realmente operativo 24 ore su 24) e milioni di utenti sparsi su tutti i continenti. In Italia i nodi attivi sono poco meno di un centinaio su tutto il territorio nazionale, una utenza valutabile, oltre le 40mila unità e in costante crescita (più del doppio delle utenze telematiche della SIP con tutti i suoi servizi sia casalinghi che relativi alle utenze d'affari, comprese anche le utenze pubbliche).
E' giusto sottolineare che un BBS è il frutto del lavoro appassionato e totalmente disinteressato di chi lo ha creato: il sysop (SYStem OPerator - operatore di sistema) che spende in questa sua creazione risorse economiche, capacità tecniche e tanto, tanto tempo.
E perché? Questo potrà sembrarvi stranissimo, ma il fine dei sysop e di coloro che li aiutano in questa opera è esclusivamente quello di permettere agli altri utenti del BBS di comunicare, di tenersi aggiornati, di divertirsi, di imparare: insomma di vivere e lavorare meglio.
E quanto costa? Proprio nulla. L'uso dei BBS Fido è totalmente gratuito per l'utente che pagherà solo la

spesa della telefonata al nodo a lui più vicino.
La regola principale per essere utenti dei sistemi Fido è quella di "non rompere troppo le scatole agli altri utenti dei BBS" e per far ciò è bene rispettare le regole che il BBS e la rete stabiliscono: nulla di trascendentale, quindi, solo buona educazione e qualche regoletta per contenere i costi alla struttura.
Un BBS è di solito diviso in due sezioni: le aree messaggi e le aree file.

Le aree Messaggi

Le aree messaggi sono delle vere e proprie discussioni interattive a cui partecipa chiunque abbia voglia di dire la propria opinione sull'argomento in questione. Tutti i messaggi presenti su di un BBS sono divisi quindi per argomento (ad es. Area Amiga, Area MS-DOS e così via) e per tipo dell'area. Quest'ultima divisione avviene perché esistono sia delle aree denominate LOCALI, che sono utilizzate solo dagli utenti di quel nodo, sia delle aree denominate ECHO, che possono coinvolgere molti nodi contemporaneamente.
Le aree ECHO possono essere limitate a pochi nodi (ad esempio, su WolfNet e su Opus Li esiste l'area PISA CONTRO LIVORNO dove l'antico campanilismo di questi due popoli si sfoga attraverso il moderno mezzo di comunicazione) oppure possono coinvolgere l'intera NAZIONE o addirittura l'intero CONTINENTE!
Semplicemente scrivendo in una AREA MESSAGGI ECHO INTERNAZIONALE nel giro di qualche giorno tutta l'Europa potrebbe leggere il vostro messaggio quindi è bene andarci cauti perché le spese gravano su tutta la struttura FidoNet. A dire il vero le aree messaggi non hanno altre regole che non siano regole di legalità di decenza e di salvaguardia dagli sprechi.
Inserire messaggi volgari od offensivi oltre ad essere poco educato, dà fastidio agli altri utenti delle aree, quindi è buona norma evitarlo.
Commerciare programmi copiati E' VIETATO DALLA LEGGE ITALIANA e se lo fate, oltre a mettere nei guai voi stessi, mettereste nei guai il vostro SysOp ed è per questo che E' TASSATIVAMENTE VIETATO FARE COMMERCIO DI PROGRAMMI coperti da copyright. Inserire un messaggio in una area ECHO genera una sequela di telefonate quasi tutte interurbane per dare la dovuta diffusione al messaggio; per questo motivo pratico E' VIETATO INSERIRE MESSAGGI NON PERTINENTI, SCORRETTI E INUTILI. A volte capita di sentirsi in dovere di ringraziare qualcuno che gentilmente ha accolto la vostra domanda e vi ha fornito una informazione: questo è giustissimo, MA NON FATELO IN UNA AREA ECHO.
Esiste una particolare area messaggi chiamata area MATRIX che serve esclusivamente alla posta personale. Il messaggio da voi immesso sarà letto solo dalla persona a cui lo indirizzate.

Le aree File

Non esistono solo i programmi commerciali. Anzi forse la parte maggiore del software prodotto è strettamente di Pubblico Dominio (PD). Il concetto di PD è un po' difficile da "digerire": il legittimo proprietario di un programma, cioè chi lo ha ideato e programmato, chiede spesso di COPIARE il suo programma a quanta più gente è possibile. Perché? Le ragioni possono essere svariate, ma comunque resta il fatto che c'è una tale quantità di software GRATIS, utilissimo e a volte pressoché perfetto e disponibile a tutti, tanto da restarne sconcertati. Le aree File dei BBS Fido servono appunto a distribuire questo software.
E' possibile, in pochi minuti eseguire un download (ovvero trasferire il programma dalla memoria di massa del BBS alla propria unità); si copiano in questo modo utility, giochi, immagini grafiche o manuali.
Nelle aree file dei vari BBS si trova di tutto, basta cercare o andare un po' a naso, oppure chiedere (le aree messaggi servono a questo, anche).
Un ultima cosa: è possibile anche mandare i vostri programmi al BBS, verranno messi insieme a tutti gli altri e chissà, dopo qualche mese un australiano vi scriverà chiedendovi notizie sulle nuove versioni del vostro programma. Ah, ricordate: più programmi mandate al vostro BBS, più ne potrete ritirare!

Altro da dire su FidoNet per questa volta non c'è, se non quello di invitarvi a provare; fate in modo di trovare un modem e collegatevi, ne sarete entusiasti, come me e come tutti coloro che lo utilizzano! ▲

Fido BBS listing as of FidoNet Nodelist for Friday, December 8, 1989

Node Board's Name	Phone Number	City	Baud
33 Italy	39-971-470547	I	2400
1Italy Coord	39-971-470547	Italy	2400
100 QuickBBS Support I	39-11-5765565	Torino	1200
101 Lynx Support Italy H	39-541-27135	Rimini (FO)	9600
102 FrontDoor Support I	39-427-938827	Italy	2400
103 D'Bridge Support I	39-766-540899	Italy	9600
104 Lynx Support Italy P	39-541-27858	Rimini (FO)	9600
105 BT Support Italy	39-6-270400	Roma	2400
106 SDN Italy	39-11-5765568	Torino	1200
331 NorthernC	39-331-263425	Cardano(VA)	2400
1 Ipotesi	39-2-2666502	Milano	2400
2 Fido Mi	39-2-33000153	Milano	2400
6 N E M O	39-2-33500529	Milano	2400
12 BBS2000	39-2-76006857	Milano	2400
14 Howard The Duck's	39-2-6551412	Milano	2400
15 Clessidra New	39-2-4159728	Milano	1200
18 HAM Link	39-2-95360079	Milano	1200
20 Fiscal Data Bank	39-2-6697700	Milano	1200
21 NEC MILANO	39-2-6697754	Milano	1200
28 The Golems' Tavern	39-2-3319638	Milano	1200
29 Opus PV	39-382-578393	Pavia	2400
31 HIGH GRAPH	39-2-24202317	Milano	9600
101 AmnesiA	39-331-263425	Cardano(VA)	2400
104 Servidati BBS	39-321-450894	Novara	2400
105 HAL bbs	39-332-286849	Varese	2400
106 SkyLink	39-332-706469	Malgesso(VA)	2400
107 TechnoNet	39-31-421391	Albese(CO)	1200
201 Amigaline	39-30-2420452	Brescia	2400
203 Euro Elettronica	39-373-86966	Crema	2400
204 Opus Bergamo	39-35-904032	Terno d'Isola(BG)	2400
332 Italy88 Telnet	39-55-605720	Italy	2400
3 SC Link	39-55-253606	Scandicci(FI)	2400
4 Digic Link	39-55-605720	Firenze	2400
5 Opus Rapallo	39-185-274020	Rapallo(SP)	2400
6 Genova 2000	39-10-3770080	Genova	2400
7 FIDO Rimini	39-541-773527	Rimini (FO)	2400
9 OCA System	39-51-6343719	Bologna	2400
10 Blue Sea BBS	39-10-3770365	Genova	2400
11 Datatel	39-59-688994	Carpi	2400
12 Utopia	39-573-368164	Pistoia	2400
13 Opus BELUSHI	39-182-51694	Albenga	2400
14 The Doors	39-574-433345	Prato	2400
15 Blues BrotherS	39-10-390287	Genova	1200
16 ARCI Computer Club	39-51-515311	Bologna	2400
18 OPUS LI	39-586-501074	Livorno	2400
19 Opus Guastalla	39-522-824379	Guastalla	1200
20 Superbit BBS	39-543-782338	Forli	2400
21 EtaBeta BBS	39-742-60063	Foligno	2400
22 San Marino Informati	39-541-774953	Rimini	9600
23 WolfNet	39-50-543180	Pisa	2400
24 Phoenix	39-544-38054	Ravenna	2400
25 AIR	39-59-396738	Modena	2400
26 AIR	39-541-777003	Rimini	2400
27 Animal House Bank	39-19-853037	Savona	2400
333 Fri-Ve-Net	39-434-32020	Italy	2400
1 Fatal Error	39-434-32020	Pordenone	2400
2 Fido Padova 1	39-49-778-018	Padova	2400
3 Fido Padova 2	39-49-620-035	Padova	2400
8 SYNAPSE BBS	39-431-438-271	Bibione	2400
12 INTUITION	39-427-938827	Sequals	2400
15 OverHead	39-49-9200-386	Padova	2400

Fido BBS listing as of FidoNet Nodelist for Friday, December 8, 1989

Node Board's Name	Phone Number	City	Baud
17 MegAmiga	39-49-630-292	Padova	2400
18 Amiga PD	39-49-8055203	Padova	2400
20 The Fake Node	Unpublished	Padova	1200
23 Treviso One	39-422-797218	Treviso	1200
101 Bit One	39-45-6860307	Verona	2400
102 Bit One 2	Unpublished	Verona	9600
103 The Wall	39-444-961708	Vicenza	2400
104 The Clivius Link	39-45-565988	Verona	2400
106 Fox BBS	39-461-821400	Trento	1200
108 Infonet Bz	39-471-200-004	Bolzano	2400
109 Time Out Bbs	39-444-504-264	Vicenza	2400
110 The NeXT BBS	39-45-31777	Verona	2400
111 MakBeth BBS	39-45-549962	Verona	2400
112 The Wolf's Hole BBS	39-474-21123	Brunico	1200
334 North-West Italy net	39-11-5765565	Italy	2400
1 Fido To	39-11-5765-565	Torino	2400
2 Opus Montecastello	39-131-355506	Alessandria	2400
3 Opus Tecnocity	39-11-4115173	Torino	2400
4 Eporedia	39-125-611624	Ivrea	2400
5 Lord Drake	39-11-710408	Torino	2400
6 Torino net #1	39-11-3100485	Torino	2400
7 Primula Rossa	39-131-42467	Alessandria	1200
8 Opus Poirino	39-11-9452705	Poirino	2400
9 EUreka!	39-11-6924400	Torino	2400
10 PC-Opus	39-11-3352858	Torino	2400
11 Charlie's Puppies	39-11-399843	Torino	2400
12 On line	39-11-9988154	Leinì	2400
13 Magazine	39-11-8180069	Torino	2400
14 E.G.O.	39-161-310585	Villata	2400
800 Travelmatic	39-11-502423	Torino	2400
335 Italia Centro Sud	39-971-470547	Italy	2400
1 Fido Pz	39-971-470547	Potenza	2400
3 Fido Roma	39-6-9035120	Roma	2400
11 List Bbs Roma	39-6-7665495	Roma	2400
15 Ennet	39-935-36174	Enna	2400
16 Roma 2000	39-6-49910315	Roma	2400
17 Joe Cocker's BBS	39-6-9031048	Roma	1200
2 S.Maria CV Bbs	39-823-812533	S.Maria	2400
4 Opus The World BBS	39-81-7433830	Napoli	2400
10 Line Bank	39-971-470547	Cassino	2400
201 The Home of Toys	39-81-7701511	Napoli	2400
6 Alex Opus	39-971-470547	Roma	1200
12 Opus Mimac Bbs	39-6-270400	Roma	2400
303EuroprogressBBS198	39-6-6232865	Roma	2400
304 Omnianet BBS	39-6-5602848	OstiaLido	2400
305EBBSG&UCourmayeur	39-6-3310165	Roma	2400
306 CED-Opus Bbs	39-6-8128656	Roma	2400
307 Mechanist's Nest	39-862-411016	L'Aquila	2400
5 Civitavecchia BBS	39-766-540899	Civitavecchia	9600
7 Civitavecchia BBS 2	39-971-470547	Civitavecchia	1200
9 DSH Soft Corp 1987	39-6-8276490	Roma	2400
405 The Best Bbs	39-6-315323	Roma	2400
406 H.H.C. Italiana Opus	39-6-8393971	Roma	1200
407 Fido Terni	39-744-402120	Terni	1200
33 Italy	39-971-470547	I	1
331 NorthernC	39-331-263425	Cardano (VA)	1
332 Italy88 Telnet	39-55-605720	Italy	2
333 Fri-Ve-Net	39-434-32020	Italy	2
334 North-West Italy net	39-11-5765565	Italy	3
335 Italia Centro Sud	39-971-470547	Italy	3

Mackie

Questo mese vi presentiamo un favoloso programma di Public Domain sconosciuto al vasto pubblico

Sebastiano Vigna

Non è infrequente, nel mondo del PD, incontrare programmi assolutamente sconosciuti al largo pubblico, ma altrettanto assolutamente favolosi. Si tratta di prodotti spesso di maggior valore delle loro controparti commerciali o addirittura non disponibili in quest'ultima veste. In particolare, ho scelto di parlare di Mackie, che sicuramente è una delle dieci migliori utility per Amiga. Scritto da Tom Rokicki (Radical Eye Software), agisce in due direzioni completamente diverse. Mette innanzitutto a disposizione un System Key simile a quello usato nelle Lisp Machine, e fornisce uno screen blanker che non ha pari nel panorama software per Amiga.

Il System Key è localizzato nel tasto di Help di Amiga. Quando viene premuta una combinazione di tasti del tipo "HELP,lettera", Mackie controlla quale programma è stato associato alla lettera attraverso il suo file di configurazione ("S:.mackierc") e provvede a portare davanti alle altre la finestra in cui il programma sta girando; se però il programma in questione non è ancora stato lanciato, Mackie lo fa partire automaticamente nel modo specificato da una seconda stringa associata al tasto. Il procedimento può essere usato sia per il software che usa la CLI sia per il software che usa solo Intuition. Ad esempio, se abbiamo deciso che la lettera "x" è associata alle stringhe "XOper" e "C:Xoper", quando noi premiamo HELP seguito da "x" Mackie controlla tutte le finestre aperte sul Workbench: se il task associato ad una finestra ha come nome "XOper", allora la finestra viene portata di fronte. Se, invece, questo non avviene, Mackie provvederà a lanciare una copia del programma mediante il comando "C:XOper". Notate che non viene controllato il titolo della window, che potrebbe tradire, bensì il nome del task associato, come risultante dalla task list. Questo sistema permette un accesso semplice, ma istantaneo ai programmi che usate più frequentemente, oppure di assegnare comandi arbitrari ad un tasto: per assicursi che vengano sempre eseguiti, basta dare una stringa improbabile, come TOSLGKJV. In ogni caso, avete a disposizione anche un comando direttamente associato ad "Amiga sinistro-Esc", nella tradizione di PopCLI e di DMouse. In un certo senso, Mackie vi permette di usare la memoria di massa come memoria virtuale per i programmi.

L'unico problema è che, una volta partito, impedirà alla CLI di chiudersi. Questo fatto si rende necessario al fine di non causare crash lanciando programmi che vogliono a tutti i costi aprire la console.device standard, e non c'è RunBack, Run >NIL: <NIL: o altro che tenga. L'unico modo per sganciare Mackie è usare "ARun NOIO" (di ARP, ovviamente) ma in questo modo il programma non ottiene una struttura di processo, bensì solo di task, quindi alla prima richiesta del System Key otterrete un Software error... In ogni caso, vi resta lo screen blanking.

E qui veniamo alla parte più coreografica del programma. Contrariamente a tanti screen blanker, che visualizzano uno schermo nero oppure al più, come Pyro, qualche fuoco d'artificio (che personalmente trovo abbastanza deprimente),

(segue a pag. 27)

Il tempo di Amiga

In quale giorno festeggeremo il nostro centesimo compleanno? Scopriamolo con questo simpaticissimo programma !

a cura di Giocologico (Tv)

Ogni società ha avuto i suoi problemi col tempo. Erano gli stregoni i "padroni del tempo" nelle società protostoriche, coloro che preannunciavano il tempo della semina e il tempo del raccolto, fornivano decotti ed eclissi. Più recentemente fu il calendario Giuliano a cercare di fornire una "sincronizzazione" accettabile alle matrone dell'Urbe e ai legionari in Gallia. Ma il nostro pianetino sembrava divertirsi a sfuggire alle regole. Per un tour intorno al sole impiega 365 giorni, cinque ore, 48 minuti ed una manciata di secondi. Col risultato di mandare in bestia chi si prova a far coincidere la data d'inizio della primavera...

L'ultimo ad intervenire per porre un po' d'ordine in materia di tempo fu Gregorio XIII che aveva il problema di "programmare" le feste di Pasqua per un po' di anni: Pasqua s'ha da tenere nella prima domenica successiva al primo novilunio di primavera. Era successo che perdendo qualche minuto ogni anno, il calendario s'era trovato indietro di 10 giorni. E così fu deciso di sopprimere dieci giorni, passando direttamente dal 4 ottobre 1582 al 15 ottobre. Inoltre, fu stabilito che da allora sarebbero stati bisestili gli anni modulo quattro ad esclusione di quelli modulo 100 tranne quelli modulo 400. Semplice no? (modulo 4 = divisione per 4 con resto zero...) Così noi, se vogliamo dotare un nostro programma di un calendario "perpetuo", dobbiamo seguire le prescrizioni di papa Gregorio. Otterremo un calendario sufficientemente accurato da poter "informare"

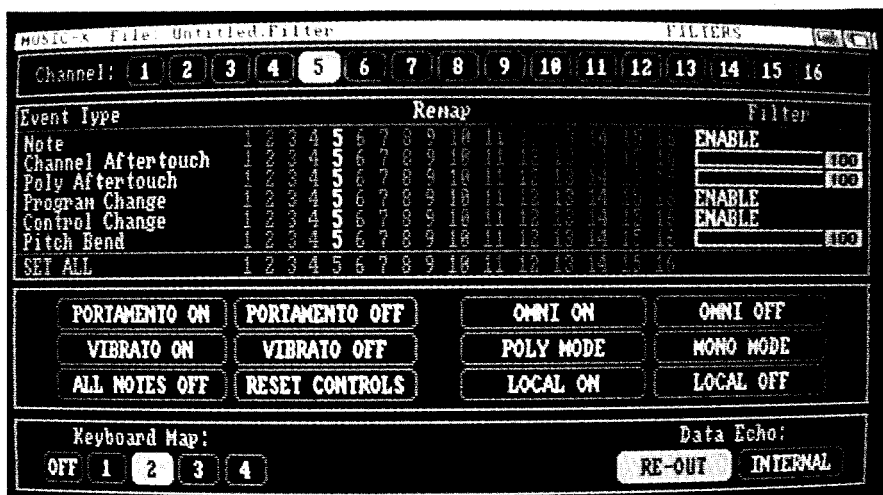
il nonno del giorno della settimana in cui è nato o in quale giorno della settimana festeggeremo il nostro centesimo compleanno anche se non potremo stabilire il giorno della settimana in cui fu scoperta l'America. Amiga misura il tempo contando i secondi dalla sua data di nascita: primo gennaio 1978. Dato che i secondi vengono accumulati in una variabile di tipo ULONG (32 bit), è evidente che la Commodore intende mantenere Amiga sul mercato per circa 4 miliardi di secondi: giusto quello che serve a noi per indagarne le possibilità. Fare i conti col tempo è un problema generale. Abbiamo perciò costruito una serie di routine che svolgano la maggior parte dei compiti di misurazione connessi al tempo. Tutte insieme, queste routine sono state impiegate per costruire un programma che possa sostituire la funzione "date" del CLI, consentendo di variare in modo più "amichevole" la data di lavoro. La nuova funzione "data" potrà essere inclusa nello startup-sequence di un disco e permetterci con qualche "clic" di aggiornare la data di lavoro. Nella maggior parte dei programmi applicativi, convertire una data in numero di giorni è conveniente perché permette di confrontare con facilità date diverse. Chi ha questo problema potrà "prelevare" le funzioni relative dalla parte bassa del programma. Chi invece ha problemi di temporizzazioni più accurate, ha a disposizione delle funzioni per consultare l'orologio di Amiga (in microsecondi....).
Nello stesso tempo le diverse routine sono un programma completo, un tipico esempio di programmazione su Amiga: finestre, gadget, images, intuitext ... un po' tutto l'armamentario di un programma scritto sfruttando Intuition. Il sorgente data.c che troverete sul disco è lungo qualcosa come 15 cartelle.
Non preoccupatevi: se togliete i commenti, di sorgente rimangono poco più di quattro cartelle.
Esistono altri modi per chiedere ad Amiga la data di lavoro: la si può chiedere ad intuition (intuitionbase), alla RootNode di AmigaDos, attraverso una funzione del lc.lib (getclk o chgclk). L'uso del timer.device è, generalmente, svincolato dal particolare tipo di compilatore che si usa o dal modo in cui si lavora ed è certamente il più interessante dal punto di vista didattico. Per chi vuole provare a "pasticciare", suggeriamo di partire da data.c per costruire con poche modifiche un "orologio" per cambiare oltre alla data di lavoro anche l'ora di lavoro. Un'ultima annotazione. Questo programma è stato compilato col Lattice 3.10, trattato con Atom in modo da mantenere i dati in memoria chip e linkato con l'Alink (alleghiamo anche il "makefile" usato: w_make). Chi usasse compilatori diversi ne tenga conto. Buon lavoro. ▲

MUSIC-X, Una potente Workstation MIDI per Amiga (Parte II)

Aldo & Andrea Laus

Bentornati al secondo appuntamento con il programma MUSIC-X. Questa volta abbiamo analizzato e provato per voi le pagine restanti del software, vale a dire: Pagina Filtri, Pagina Amiga Samples e Pagina Librarian.
Per quanto riguarda, invece, la Pagina Sequencer facciamo riferimento alla prima puntata, apparsa sullo scorso numero di Amiga Magazine.

Pagina filtri

Come abbiamo detto la volta scorsa, è forse la pagina più interessante del programma e, comunque, vale, da sola, il prezzo dell'intero pacchetto.
Si raggiunge da una qualsiasi delle altre tre pagine del MUSIC-X e viene usata per processare i dati MIDI in ingresso, prima che vengano utilizzati dalle altre pagine.
Dato che esiste la funzione ECHO, che rende disponibili su MIDI OUT gli stessi dati, ne deriva che, senza necessariamente fare funzionare il sequencer, possiamo utilizzare queste potenti funzioni anche per gestire complesse reti MIDI in tempo reale. E non crediate che per rendere complessa una rete MIDI serva poi molto!
Chi dispone già di almeno un synth, un expander multicanale, una drum machine e un sequencer, secondo noi, ne ha già bisogno. Vediamo come si presentano nell'insieme le prestazioni delle pagine Filtri e Keymap: i filtri di cui dispone MUSIC-X sono in grado di trasformare i messaggi di nota in altri tipi di evento, impedire il passaggio, per risparmiare memoria, ad alcuni tipi di messaggi, "filtrare" la quantità di messaggi selezionati, cambiare il canale MIDI dei tipi di eventi selezionati e ri-dirigere i messaggi alle voci interne di Amiga o all'output MIDI.
Queste sono le più comuni operazioni normalmente svolte da un filtro MIDI.
Nel MUSIC-X ci sono sedici filtri completi, uno per ogni canale MIDI, che funzionano contemporaneamente.
La grande utilità di questa pagina è quella di interpretare canali MIDI multipli in tempo reale, come, ad esempio, durante la registrazione di dati dal "vivo" di un gruppo di strumenti MIDI attraverso un Merge Box.
La trasformazione di eventi di nota in altri tipi di evento si effettua usando le Keymap.
Una Keymap permette un'interpretazione diversa di ogni tasto su una tastiera MIDI.
Il programma può tenere in memoria fino a quattro diverse keymap che sono, in sostanza, dei Patche.
Si possono mdificare le quattro Keymap correnti del MUSIC-X tramite il modulo Keymap Editor; il programma vi consente di salvarle su disco, permettendovi così di poter creare una libreria.
La Pagina Filtri è divisa in quattro finestre. La prima finestra in alto contiene una fila di icone, numerate da 1 a 16, che rappresentano il canale MIDI e servono a determinare il filtro corrente.
La seconda finestra contiene i controlli del filtro veri e propri, usati per filtrare o ignorare i tipi di evento selezionati; qui viene effettuata la ricanalizzazione dei dati.
La terza finestra contiene una serie di icone relative a messaggi MIDI, quali Portamento on/off, Vibrato on/off, All note off, Omni on/off, Poly on/off, Local on/off, che possono essere inviati ai dispositivi, sul canale corrente.
Questi non sono esattamente dei controlli dei filtri, ma possono essere usati per configurare o inizializzare i synth collegati al computer.

La quarta finestra contiene due tipi di informazioni: a sinistra ci sono i selettori per scegliere con quale delle quattro Keymap lavorare; a destra ci sono i selettori Data Echo che determinano se i dati, sul canale operativo, dopo essere stati filtrati, vengono diretti all'uscita MIDI o ai suoni interni di Amiga o se non saranno diretti per niente.
I sei tipi di eventi MIDI in entrata che possono essere ri-canalizzati sono: Note, Aftertouch di canale, Poly Aftertouch, Program Change, Control Change, Pitch Bend.
Interessante notare che tutti e sedici i filtri possono essere salvati in un file Performance nella pagina Sequencer.
Tramite il menu dei moduli si accede alla sotto-pagina Keymap Editor. Una Keymap usa gli eventi MIDI di nota per provocare altre azioni.
Queste comprendono la possibilità di suonare degli eventi MIDI, attivare gli eventi del MUSIC-X e farne partire i comandi.
In sostanza, una Keyboard Map rappresenta un elenco dei 128 numeri di nota MIDI in ingresso e un corrispondente elenco delle azioni che si verificheranno quando ognuno di questi tasti viene premuto.
Nella maggior parte dei casi, ci saranno gruppi di tasti che provocano lo stesso tipo di azione. Per esempio, quando si usa una Keymap per splittare una tastiera in due, per suonare con due synth diversi, molti dei tasti, se non tutti, dovrebbero provocare la stessa azione: quella di produrre eventi di nota.
A tal scopo viene definito un Range, consistente di un gruppo di tasti adiacenti, regolati in modo da provocare la stessa azione.
Un Range può comprendere l'intera tastiera MIDI oppure un solo tasto e si può definire un numero a piacere di Range.
Il Keymap Editor viene usato per modificare le quattro Keymap usate nella Pagina Filtri.
I controlli di questo modulo sono divisi in quattro aree: in alto c'è il pannello dei Modi, per scegliere l'attuale mappa e per prendere decisioni globali di edit; la seconda area contiene la tastiera in miniatura, usata per la creazione e l'editing dei range di note sulla mappa attuale; in basso c'è l'elenco delle azioni e la finestra dei parametri, usata per scegliere le azioni da assegnare ad ogni range. Nell'editing di una Keymap ci sono quattro passi basilari da seguire: viene scelta una delle quattro keymap da editare; si definisce un range di note e, se necessaria, la relativa trasposizione; si definisce l'azione che verrà provocata dai tasti di quel range e si fissa, sulla memoria del computer o su disco, la keymap editata e pronta all'uso.
La Figura 1 visualizza i possibili percorsi del flusso dei dati durante la registrazione, quando vengono usati i filtri e le keymap.
I dati provenienti dalla vostra tastiera musicale vengono ricevuti tramite la presa MIDI IN del computer; i dati vengono divisi in due flussi: gli eventi relativi alle note seguono una direzione; tutto il resto va ai filtri.
Secondo lo stato del selettore keymap, le note vengono inviate al modulo keymap, oppure ai filtri assieme agli altri dati.
Le note inviate al keymap vengono trasformate in altri eventi.
Gli eventi MIDI che vengono prodotti, vengono trasmessi attraverso il selettore DATA ECHO, mentre gli eventi Play vengono "presi" dal Record Buffer; se il selettore Re-out è attivo, vengono ritrasmessi verso il MIDI OUT del computer, consentendovi di sentirne l'effetto sui vostri strumenti collegati.
Le sequenze che stanno suonando durante la fase di registrazione, vengono processate attraverso il canalizzatore e ritrasmesse al MIDI OUT.
Nel caso in cui l'opzione MIX-DOWN-MODE è attiva, i dati delle sequenze vengono anch'esse prese da Record Buffer.
Alla fine, esso potrebbe contenere dati alquanto differenti, rispetto a quanto è stato suonato sulla vostra tastiera, in funzione di ciò che avete predisposto nelle Pagine Filtri e Keymap.

La Pagina Amiga Samples

Questa pagina è dedicata alla gestione dei suoni campionati utilizzabili con i quattro generatori interni di Amiga.
Si possono caricare in memoria da disco fino ad un massimo di sedici suoni contemporaneamente, sia in formato IFF che Sonix.

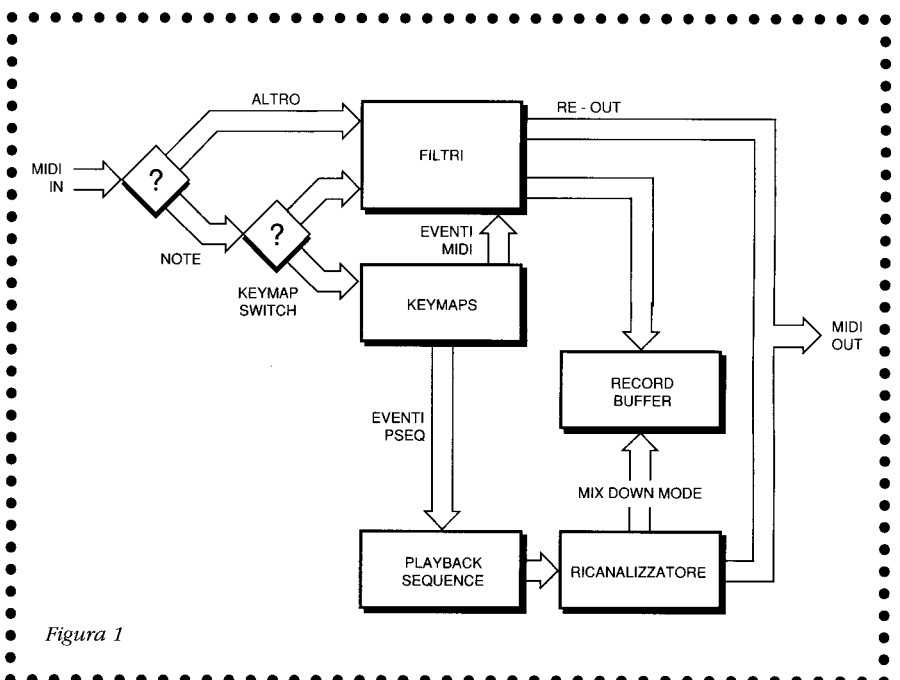
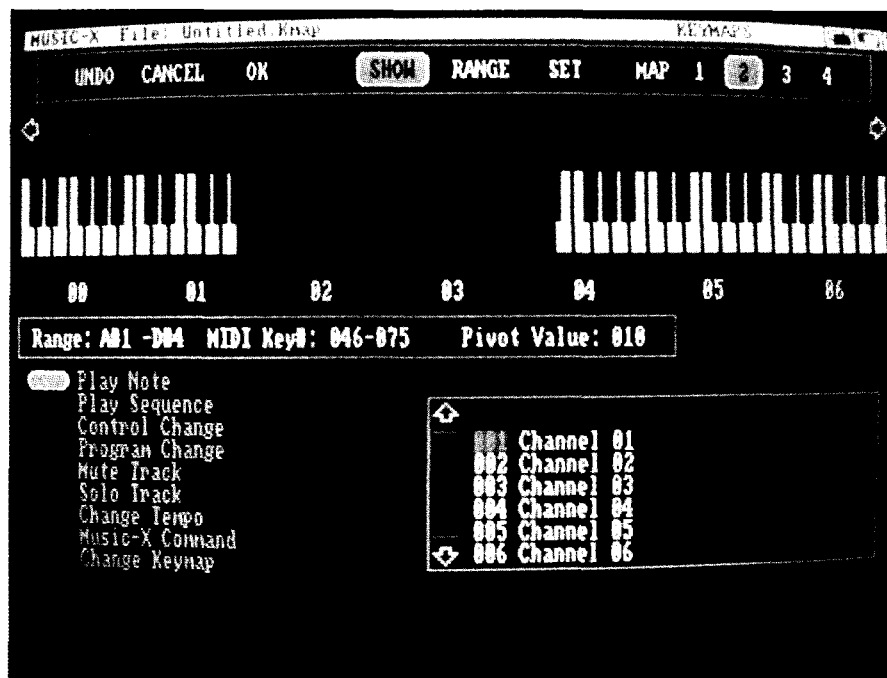
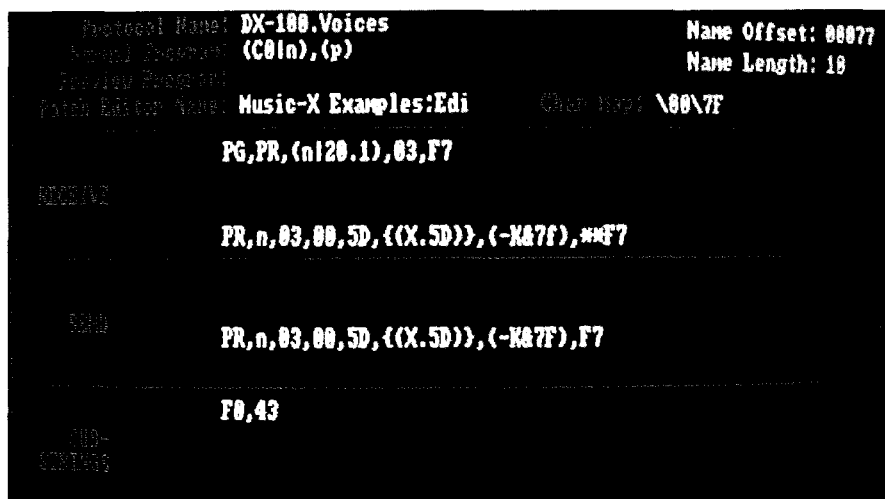



Figura 1

La polifonia di Amiga è limitata a quattro voci e, quindi, sono solo quattro i suoni utilizzabili contemporaneamente, dal sequencer o, in tempo reale, da una tastiera esterna collegata via MIDI.
Utile l'opzione offerta da questa pagina di poter editare graficamente l'inviluppo dei singoli campioni *fino a sedici punti di edit; si può, in*oltre, modificarne l'accordatura, il range e scegliere se attivare o meno il Filtro di Amiga.
La nostra opinione circa questa pagina è, nel complesso, positiva, nonostante il fatto che la qualità dei campioni non è paragonabile a quella dei suoni prodotti da campionatori professionali, il suo utilizzo può tornare utile a coloro che, non avendo magari ancora acquistato un campionatore o trovandosi nell'impossibilità di collegarsi via MIDI con altri expander, vogliono comunque avere a disposizione dei suoni personalizzati, per quanto "rozzi" *possano essere, con cui lavorare.*
Può, inoltre, capitare che ci si trovi nella condizione di dover incidere, soprattutto nei moderni "Mix" da discoteca, delle parole o delle piccole frasi all'interno del nostro pezzo. A questo punto, perché sprecare uno, se non di più, dei pochi e preziosi canali del nostro campionatore, quando ci possiamo servire comodamente dei campioni Amiga?
Importante notare che la Pagina Samples non permette né l'incisione dei campioni né un loro edit completo; per questa funzione bisogna affidarsi a uno dei tanti programmi *dedicati, che può essere caricato* con Music-X in multitasking.

La Pagina Librarian

Costituisce uno dei punti di forza dell'intero pacchetto. Serve per immagazzinare in librerie residenti su dischetti Amiga i dati dei suoni delle vostre tastiere.
Questo sistema permette di caricare in memoria e successivamente scaricare nelle RAM degli expander collegati dei banchi di suoni. Il vantaggio di questa pagina è duplice: prima di tutto consente al musicista *di risparmiarsi* l'acquisto di onerose cartucce su cui scaricare i suoni; in secondo luogo, e qui sta la sua forza, consente di programmare ed editare i protocolli, per realizzare il dump di Sistema Esclusivo.
I normali programmi Librarian, facenti parte, di solito, di un unico software comprendente anche un editor dei suoni, sono "dedicati", elaborati cioè per interagire con un unico sistema esclusivo, quello del synth di cui editano e immagazzinano i suoni. Se un musicista possiede più sintetizzatori, è costretto a comprarsi più editor, uno per ogni synth. La pagina Librarian del Music-X *permette, invece, di editare i proto*colli, in modo da poter interagire con più synth diversi fra loro. Sul terzo dischetto, quello degli esempi, sono presenti alcuni fra i più diffusi protocolli pronti all'uso (ad esempio quelli per DX7, D50, MT32, Matrix6...). Dobbiamo però osservare che lo sfortunato musicista che possedesse una tastiera poco nota, il cui protocollo non è presente tra quelli offerti, dovrà darsi da fare per crearsi il proprio protocollo. Per riuscire nell'impresa, dovrà possedere, come minimo, un po' di dimesti-

```
DX-100.Voices                 Name Offset: 00077
(C0In),(p)                    Name Length: 10
Music-X Examples:Edi          \00\7F
PG,PR,(nI20.1),03,F7
PR,n,03,00,5D,((X.5D)),(-K&7f),**F7
PR,n,03,00,5D,((X.5D)),(-K&7F),F7
F0,43
```

chezza con il codice esadecimale e il protocollo MIDI.
Agli utenti senza queste caratteristiche, il manuale consiglia di attendere ulteriori aggiornamenti. In ogni caso gli interessati possono contattare la Soundware che provvederà a fornire loro un fascicolo che tratta la creazione dei protocolli. Sul dischetto Esempi sono presenti, a titolo dimostrativo, anche patche di suoni per le tastiere più diffuse. Dalla pagina Librarian si accede ai moduli Editor per l'edit dei suoni immagazzinati (vedere anche la figura sulla puntata precedente).
I Patch editor grafici, in particolare quello per il D50, sono, nel complesso, ben studiati, mentre il Generic Patch editor, che si presenta sotto forma di semplice listato di dati, lascia un po' a desiderare nella comodità di utilizzo. E', secondo noi, una lacuna il fatto che il manuale accenni brevemente ai soli Generic Editor, non facendo parola di quelli dedicati.

Conclusione

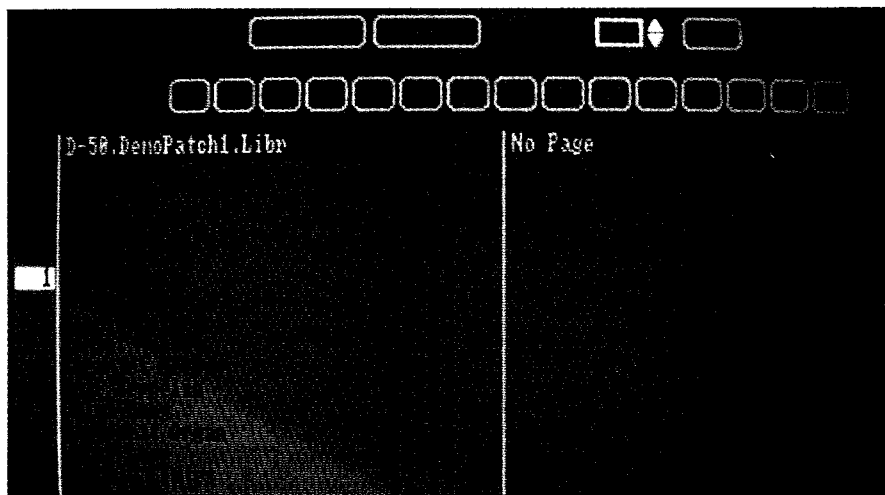
Il pacchetto Music-X, che abbiamo provato con un Amiga 2000, ci sembra uno fra i migliori software musicali oggi disponibili per Amiga. Offre, infatti, tutte le prestazioni di cui un musicista può necessitare, mantenendo una struttura compatta. Importante ricordare che si tratta di uno dei pochi software del genere, sviluppato e pensato per Amiga, e non una trasposizione da altri computer.
Date le particolari possibilità che offre, è destinato, in particolare, ad una schiera di professionisti che, certamente, ne apprezzeranno le potenzialità. Per il dilettante o per chi non ha intenzione di programmare delle Keymap o di lavorare con gli editor dei suoni, Music-X rappresenta comunque un buon investimento, considerato l'ottimo sequencer di cui dispone e la possibilità di lavorare direttamente con i suoni interni di Amiga. Music-X rappresenta, secondo noi, il programma tipo verso cui, in futuro, tutte le software house si orienteranno: è una workstation completa che mette a disposizione del musicista tutti i mezzi per lavorare. E' in studio, inoltre, un nuovo modulo Score Writer, con il quale il pacchetto Music-X diventerà davvero completo.
Un altro motivo per acquistare questo software dal costo contenuto (circa £. 399.000 IVA inclusa) in rapporto a ciò che offre, è dato dalla notevole assistenza (di cui abbiamo parlato nella precedente puntata) offerta dalla ditta Soundware, che distribuisce il programma in Italia. ▲

Per ulteriori informazioni, rivolgersi a:

Soundware S.r.l.
Via Mazzini, 12
21020 Casciago (VA)
Tel: 0332-222052

(segue da pag. 21)

Mackie

Mackie sfoggia uno schermo ad alta risoluzione da 1 bitplane (circa 20K di Chip RAM) su cui fa danzare delle meravigliose spline, ovvero sine polinomial line, linee generate mediante polinomi di seni. Tralascio la descrizione, che sarebbe assolutamente inadeguata, e vi consiglio semplicemente di lanciarlo con "Run Mackie 1", comando che attiverà, dopo un secondo, lo schermo del blanker. L'impressione di animazione è data dal fatto che Mackie disegna un certo numero di spline, dopodiché ogni volta che una viene aggiunta, la prima viene cancellata. L'algoritmo, come potete vedere dal sorgente Manx, è per buona parte in linguaggio macchina. Tanto per fare un paragone rispetto a DLineArt, lo screen blanker esterno per DMouse che visualizza qualcosa di simile, basti dire che il codice di DLineArt è stato preso di peso da Mackie, come dichiara lo stesso autore. Inoltre, Mackie ha il buongusto di mettersi a priorità -20: grazie a questo fatto, eventuali programmi che stiano girando (ad esempio un compilatore) non verranno praticamente rallentati, in quanto di fronte ad una loro richiesta di tempo CPU, Mackie deve necessariamente farsi indietro. Una possibilità d'uso non documentata del programma è in effetti quella di monitorare lo sfruttamento che un altro programma fa del 68000. Ogni volta che Mackie si ferma o rallenta c'è un uso proporzionalmente intensivo della CPU.
Tom aggiunge continuamente opzioni, tra cui l'eliminazione delle spline (anche solo temporaneamente) mediante "Mackie -b". In ogni caso, il programma può essere rimosso in qualunque momento con "Mackie -q". Notate che dopo la prima invocazione, necessariamente preceduta da Run, Mackie può essere lanciato da solo, in quanto si limita a comunicare alla copia già presente in memoria eventuali nuovi parametri. Non vi resta che inserirlo nella startup-sequence... ▲

Parola ad Amiga

Il nostro meraviglioso Amiga riesce anche a parlare !! Scopriamo assieme, in modo molto semplice, come.

Gianni Biagini

Una caratteristica di AMIGA, che non è supportata dalla maggior parte degli altri computer e che attrae per la sua natura, è la sintesi vocale: Amiga, infatti, possiede una struttura software e una serie di librerie che ne gestiscono la voce.
Come fa un computer a sapere come pronunciare le lettere che compongono una frase? Come fa a determinare quale intonazione o accento deve dare ad ogni componente del discorso?
Per prima cosa vediamo di determinare il concetto di ALLOFONO, ossia di componente fonetica composta da un gruppo contiguo di lettere dell'alfabeto.
Pensate alla frase inglese "I am an AMIGA", provate a leggerla secondo la fonetica italiana, sicuramente un inglese non capirebbe quanto proferite.
Il ragionamento, naturalmente, si può fare anche con il francese "je suis un Amiga"; già con il tedesco le differenze diminuiscono, poiché la lingua di cui sopra è foneticamente abbastanza simile all'italiano (cosa che non è assolutamente vera per i vocaboli) e quindi "Ich bin ein Amiga" letto da un italiano può essere facilmente capito da un tedesco.
Cosa abbiamo dedotto?
Che lettere uguali possono essere pronunciate diversamente se lette da persone parlanti lingue diverse.
Pensate al vostro vecchio libro di inglese delle medie: ricordate che sotto alle parole c'erano degli strani segni?
Essi rappresentavano la pronuncia fonetica delle parole e una volta acquisito il suono ad essi associato era possibile leggere ogni parola senza biunivocità.
Come si può allora far leggere ad Amiga una frase facendo in modo che la interpreti secondo la fonetica inglese, italiana o giapponese? Attraverso gli allofoni, naturalmente!
Intendiamoci, non potete inserire da tastiera di Amiga gli allofoni tipici (quelli detti I.P.A. ossia International Phonetic Alphabet) poiché non ci sono i tasti predefiniti, però attraverso una serie di comuni caratteri si possono scrivere tutti i codici dell'ARPABETO, il sistema alternativo di scrittura fonetica messo a punto dalla Advanced Research Projects Agency, l'ARPA, per intenderci.
Il linguaggio AmigaBasic mette a disposizione una funzione che, data una stringa in ingresso, la traduce nei rispettivi segni dell' ARPAbeto: vediamo il seguente listato

```
a$= "Hallo people, how are you?"

b$= TRANSLATE$(a$)
```

un messaggio viene assegnato alla variabile A$ e viene scritto in codice ARPAbetico nella variabile B$;

```
SAY b$
```

infine fa dire al computer il messaggio. Abbiamo finalmente insegnato ad Amiga i primi passi della comunicazione orale; vediamo ora, per spirito nazionalistico se volete, di studiare un sistema per far parlare Amiga in italiano (dato che TRANSLATE$ traduce in allofono che danno la pronuncia inglese della frase).
Pensiamo a qualche tipica parola inglese, per esempio TREE (albero).

Ci accorgiamo che la doppia E si legge I, è un buon passo avanti poiché TRANSLATE$ l'avrebbe letto AI !!
Continuiamo, quindi, a cercare parole inglesi che ci facciano risalire alla pronuncia italiana, BUT si legge BAT però PURE si legge PIUR, troppe ambiguità, andiamo oltre. Pensate a ROOM, LOOM, ZOOM (rispettivamente stanza, telaio e lo zoom delle macchine fotografiche), la doppia O si legge U, abbiamo un'altra piccola tesserina da aggiungere al nostro mosaico, ricordando peraltro che le consonanti inglesi non si discostano particolarmente da quelle italiane, la nostra ricerca si limita principalmente alle vocali e si prepara una tabellina di corrispondenza così da poter poi sentire il nostro Amiga parlare la nostra lingua.
Vi sono due possibili soluzioni per completare la nostra opera, una è quella di pensare a cosa dovrebbe leggere un inglese per usare la fonetica italiana. Mi spiego: perché un inglese dica IO deve leggere EEO poiché, come già detto sopra, la doppia E viene letta I.
Vediamo ora più in dettaglio quale sintassi accetta il Basic di Amiga per parlare: innanzitutto abbiamo parlato della TRANSLATE$ che converte la stringa in segni ARPAbetici, vediamo ora come assegnare i parametri caratteristici alla voce vera e propria; il comando per far parlare Amiga è SAY "stringa", vettore di modo.
Per la stringa non ci sono problemi, possiamo scriverla con i segni ARPAbetici o "passarla" alla funzione attraverso la sopraccitata TRANSLATE$:

```
a$="I am an Amiga speaking"

say(translate$(a$))
```

Il vettore di modo, invece, dà alla funzione una serie di importanti dati quali:

elemento 0: PITCH è l'intonazione della voce, il parametro va da 65 a 320 dove 110 è un valore normale

Valore	Canale
0	0
1	1
2	2
3	3
4	0 e 1
5	0 e 2
6	3 e 1
7	3 e 2
8	Qualsiasi canale sinistro disponibile
9	Qualsiasi canale destro disponibile
10	Qualsiasi canale di destra e qualsiasi di sinistra
11	Qualsiasi canale singolo libero

Tabella 1

elemento 1: INFLESSIONE può assumere due valori, 1 inflessione robotica o 0 inflessione umana

elemento 2: VELOCITA' può assumere i valori da 40 a 400 (parole al minuto)

elemento 3: SESSO può essere maschile (0) o femminile (1)

elemento 4: TONO esprime la frequenza della voce, va da un minimo di 5000 ad un massimo di 28000, (voce acida e squillante)

elemento 5: VOLUME da 0 (silenzio) a 64

elemento 6: CANALE si può scegliere a quale dei quattro canali stereo di Amiga far dire il messaggio; vedere tabella 1.

elemento 7: MODO può essere sincrono ossia Amiga aspetta di finire di parlare prima di continuare con il programma (0) o il contrario (1)

elemento 8: CONTROLLO dice ad Amiga come comportarsi in caso di interferenza tra più SAY:
0 = Amiga finisce la SAY corrente e poi passa alla successiva
1 = Amiga interrompe ciò che stava dicendo e non processa la SAY seguente
2 = Amiga interrompe la vecchia SAY per passare subito a quella nuova

Ricordate che il vettore deve essere composto da variabili intere, quindi assegnategli un nome che finisca con il segno '%' vedi ad esempio MODO%

Ricapitoliamo: TRANSLATE$ traduce una stringa in segni ARPAbetici, SAY la dice secondo delle modalità espresse nel vettore di modo, quindi un semplice programmino potrebbe essere il seguente:

```
FOR N=1 to 8
  READ modo%
' leggiamo i dati in fondo al
programma
NEXT N
```

SAY TRANSLATE$ ("THIS IS AMIGA MAGAZINE, THE BEST FOR YOU")
Amiga dirà ora la frase traducendola prima in ARPAbeto DATA 120,0,145,0,22200,64,10,0,0

Per un ulteriore esempio della programmazione in Basic riguardante la sintesi vocale si rimanda il lettore al programma incluso nel disco EXTRAS e dal nome SPEECH. ▲

MGM: Realizzazioni grafiche su richiesta

Da oggi è possibile ordinare immagini, animazioni, digitalizzazioni e slide-show di qualsiasi tipo sia su disco che su videocassetta ad un costo relativamente basso....

Daniele Cassanelli

Grazie alle potenti caratteristiche grafiche di Amiga e ai validi programmi che ne sfruttano al massimo le potenzialità, si è giunti ad un livello tale, nella produzione video, che sono nate ditte specializzate per la realizzazione di animazioni e disegni grafici, "su misura", con Amiga. Il motivo principale per cui nascono le suddette ditte è la necessità, da parte di enti o di privati, di creare degli slide-show personalizzati, per pubblicità o per diletto, a basso costo. Infatti, per la creazione di buone animazioni, è necessario aggiungere ad un Amiga base dispositivi che permettono di aumentarne le potenzialità grafiche, che per ditte o per privati, hanno un costo inaccessibile.

MGM Studio

La MGM Studio è una piccola ditta di Vobarno (BS) che realizza appunto prodotti grafici ad un tale livello che tra i suoi clienti, oltre ai privati, sono presenti anche emittenti televisive. Questa ditta utilizza Amiga 2000 equipaggiati con schede acceleratrici fornite di 68030 e 68882 a 25 MHz, Hard Disk SCSI da oltre 100 Mb, espansioni da 8 Mb e digitalizzatori, inoltre, utilizzano software grafico aggiornato alle ultime novità. La MGM, dove il nome, come qualcuno potrebbe sospettare, non sta per "Metro Goldwin Mayer" ma deriva dalle iniziali dei proprietari, ci ha gentilmente fornito del materiale dimostrativo da visionare: nei dischi troviamo, oltre ad immagini digitalizzate e in 3D, anche animazioni e slide-show di notevole livello, per non parlare delle stupende immagini in ray-tracing che lasciano allibiti.
Le immagini 3D sono da visionare obbligatoriamente con gli occhiali 3D (con una "lente" in plastica rossa e l'altra in plastica verde) che la stessa MGM ci ha fornito. I prezzi non sono molto elevati (visto l'ottimo livello dei loro prodotti), tra cui: disco con animazione a 15.000 lire (è possibile applicare il metodo prendi tre, paghi due), disco con immagini

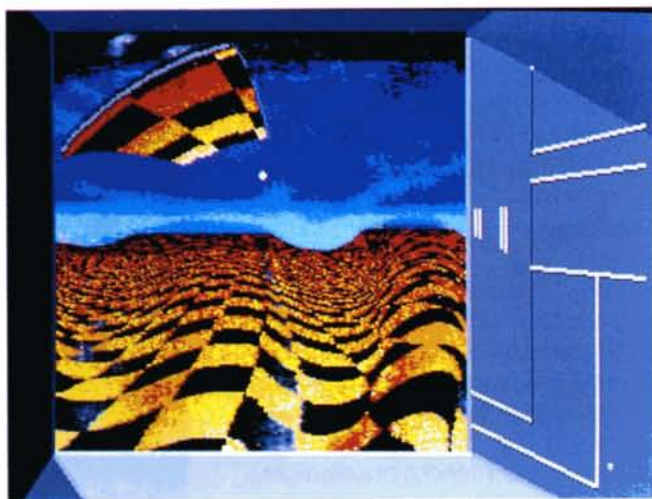
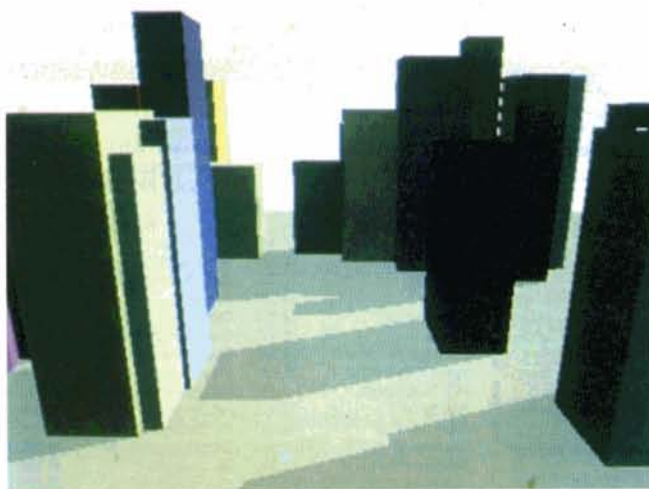
digitalizzate e in ray-tracing a 10.000 lire l'uno e videocassetta VHS con animazioni e digitalizzazioni da 50.000 lire. Non mancano, naturalmente, immagini sexy e "nudi d'autore" (come li chiamano loro).

Conclusioni

Se non volete spendere elevate cifre per ottenere una pubblicità o una presentazione accattivante in computer grafica realizzata con una potente WorkStation, potete avere risultati molto simili ad un costo decisamente inferiore grazie all'aiuto di queste ditte più accessibili.

Per ulteriori informazioni contattare: M.G.M. Studio Via G.Verdi, 6 - Vorbano (BS) Tel.0365/598757

Amiga Art Gallery è una rubrica mensile aperta a tutti gli Amigartisti.
Per partecipare è sufficiente inviare una fotografia, diapositiva o dischetto, con l'immagine salvata in formato IFF, della propria opera con qualche riga di descrizione come: nome dell'autore, programma con cui avete creato l'opera, oppure il metodo utilizzato ecc.
Ricordiamo però che Art Gallery non è né una gara né una competizione !!
Le opere pubblicate vengono scelte a giudizio insindacabile della redazione, le opere pervenute non vengono restituite.
Se veramente vi ritenete degli artisti con Amiga, cosa aspettate a inviarci le vostre opere, magari scopriamo un nuovo Picasso !!

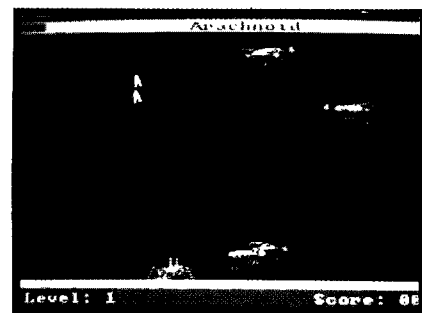
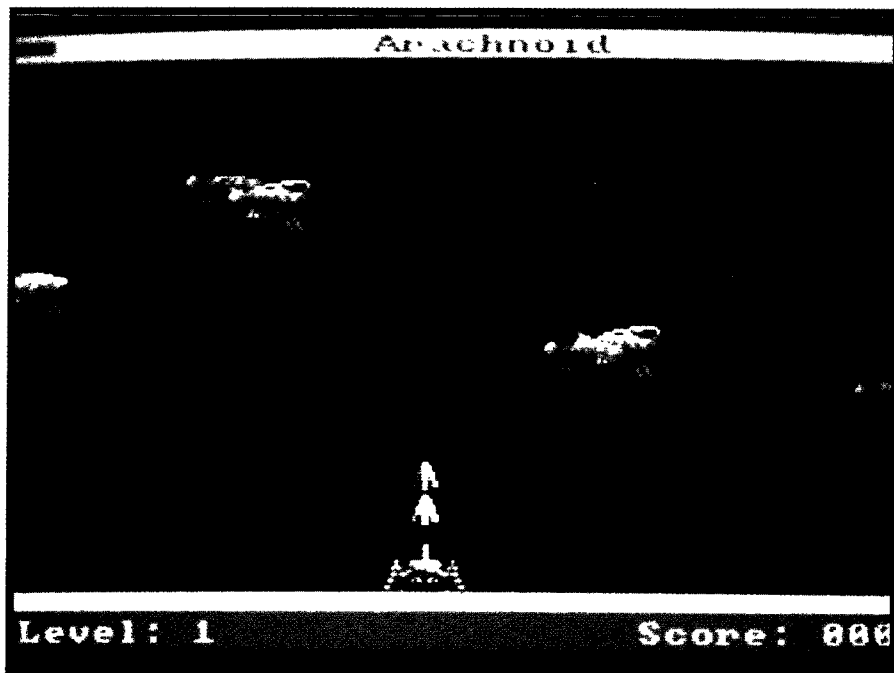
Le opere vanno inviate a:
Gruppo Editoriale Jackson
Amiga Magazine
Area Consumer
Rubrica "Art Gallery"
Via Pola, 9
20124 Milano

Camera con vista
Paolo Brivio
Programma: Deluxe Paint III

Città del 2000
Alessandro Turra
Programma: Digi Paint III

Pareti di un labirinto
Erica Croci
Programma: Deluxe Paint II

ON DISK è una rubrica mensile di quattro pagine che possono anche essere staccate e conservate, in queste pagine sono descritte tutte le informazioni dei programmi inclusi nel disco, complete di istruzioni, trucchi ecc... In questo spazio troveranno posto giochi, utility e tutto ciò che può fare Amiga.

• Games

Arachnoid

Lo scopo di Arachnoid consiste nel neutralizzare, con l'ausilio di un ragno meccanico fornito di un missile riutilizzabile, nove ondate di vespe giganti. Quando il gioco inizia, la prima ondata di vespe è già all'interno dello schermo e, tramite il joystick potete muovere l'Arachnoid (destra, sinistra) farlo saltare (alto) e lanciare il missile. Per atterrare (senza però neutralizzare) una vespa è sufficiente colpirla con il missile; la vespa colpita cadrà al suolo, ma dopo alcuni secondi riprenderà il volo ancor più agguerrita. Per neutralizzare definitivamente una vespa atterrata, è necessario iniettarle del veleno che, però, all'inizio dell'ondata, non è a vostra disposizione. Per venirne in possesso è necessario toccare uno dei bonus che appaiono vicino al soffitto; ma solo saltando non si può arrivare al soffitto; è allora necessario, mentre si è in volo, portare la leva del joystick verso il basso: l'Arachnoid si capovolgerà e grazie alla molla che spara il missile, con due o tre salti arriverà al soffitto dove potrà attaccarsi e camminare proprio come sul pavimento per andare a prendere il veleno. Da quel momento in poi, portandosi sopra una vespa atterrata (tornando sul pavimento e girandosi in volo per non rimbalzare) e muovendo la leva del joystick verso il basso, sarà possibile neutralizzare completamente la vespa. E' importante, durante tutte queste operazioni, tentare di farsi toccare il meno possibile dalle vespe, perché dopo un certo numero di punture il gioco termina.

Alla fine del gioco se si è totalizzato un record, è possibile inserire il proprio nome nella high score. Vi consigliamo in qualsiasi caso di lasciare protetto il disco allegato alla rivista oppure di farne una copia personale e lasciarla sprotetta durante il gioco; tutto questo perché i virus sono sempre in agguato !!

• Utility

V - Versione 2

Nel disco allegato al numero 7 di Amiga Magazine è stato pubblicato V, un programma per visualizzare facilmente immagini di qualsiasi formato; in questo numero vi presentiamo una versione migliorata di questa utility; tra le migliorie apportate al programma c'è la possibilità di creare degli slideshow scrivendo un file batch grazie all'opzione di temporizzazione; inoltre è ora possibile utilizzare per l'identificazione dell'immagine, da visualizzare, delle wildcard.

Come creare uno slideshow

Per creare uno slideshow è sufficiente scrivere un file batch AmigaDos (tramite, ad esempio, l'editor ED presente nel WorkBench) contenente tutte le chiamate a V con i relativi file-immagine e il tempo che deve trascorrere dal caricamento di una figura e della successiva, nel seguente modo:

```
V AMIGA.PIC MOUSE.PIC 10
V DRIVE.PIC 20
V PICTURES/~*.info 5
```

Una volta salvato il file batch sarà possibile mandarlo in esecuzione digitando da CLI EXECUTE nomefilebatch.
Questo particolare file batch visualizza le due figure AMIGA.PIC e MOUSE.PIC lasciandole sullo schermo per 10 secondi ciascuna, successivamente visualizza DRIVE.PIC lasciandola sullo schermo per 20 secondi. L'ultima linea mostra l'uso delle wildcard, che sono le seguenti:

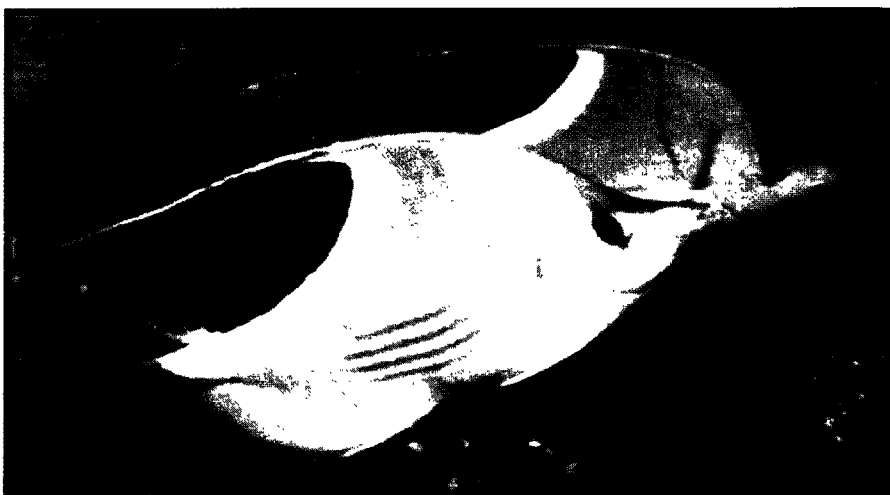
" * " serve ad indicare che a partire da quella posizione può esserci una qualsiasi sequenza di caratteri di una qualsiasi lunghezza.
Un esempio dei possibili nomi di file individuati da FOR*.PIC potrebbe essere rappresentato dai nomi di file FORTUNATO.PIC, FORZIERE.PIC, FORMULA.PIC eccetera.

" ~ " serve, invece, a negare il significato del nome del file contenente wildcard in modo da escludere (invece che includere) tutti i nomi di file che lo ripettano; ecco un esempio:
Nome del file indicato con wildcard: ~PAL*.PIC
Non verranno visualizzati tutti i file il cui nome inizia per PAL.
Nell'esempio riportato nel file batch verranno visualizzati tutti i file contenuti nella directory PICTURE che hanno dopo il punto una parola diversa da INFO.
Un'altra innovazione apportata a V consiste nella possibilità di visualizzare immagini nel formato SHAM (Sliced HAM), immagini cioè che modificano ad ogni riga dello schermo i colori di base permettendo di ottenere immagini prive dell'effetto tipico dell'HAM causato dal fatto che non è possibile modificare nel passaggio da un pixel al successivo più di una componente cromatica (ROSSO, VERDE o BLU).

Inoltre, V supporta ora anche il Color Cycling tipico delle immagini del Deluxe Paint; premendo il tasto TAB durante la visualizzazione è possibile attivare il Color Cycling.
Per chi non avesse acquistato la rivista in cui è stata pubblicata la prima versione di V verranno ora spiegate le caratteristiche del programma comuni anche alla prima versione.
V supporta i seguenti formati IFF (oltre allo SHAM): bassa o alta risoluzione (320 o 640 pixel orizzontali), interlacciato o non-interlacciato (256 o 512 pixel verticali), Hold And Modify (HAM) interlacciato e non, Extra Half Brite (EHB).
V permette, inoltre, di visualizzare Brush e immagini che sfruttano l'overscan.

Per utilizzarlo vi sono vari modi:

1) Tramite CLI dando il comando :
V <nomefigura>

2) Da WorkBench, cliccando sull'icona della figura da guardare, tenendo premuto il tasto SHIFT della tastiera e cliccando due volte sull'icona di "V" (metodo delle selezioni multiple).
Se volete osservare più di una immagine potete farlo cliccando normalmente sulla prima, cliccando

ViewPort

Un file requester di sistema... o no?

Larry Phillips

"Vorrei del gelato alla fragola, per favore"
"Spiacente, abbiamo solo vaniglia"
"Nessun'altro gusto?"
"No, ma la vaniglia è il gusto migliore. Senza alcun dubbio, il solo gusto di cui lei abbia bisogno. E' freddo, di un grazioso colore bianco-giallognolo e non offende il palato come fanno certi altri sapori. E, soprattutto, rende la scelta molto semplice"
"Ma se non mi piace la vaniglia?"
"A tutti piace la vaniglia. E' per questo che la serviamo. Se veramente non le piace, può sempre farsi il suo gelato con la nostra comoda macchina per gelati."

Gelato in una rivista sui computer? Il fatto è, voi direte, che Phillips è veramente rimbambito!
La conversazione non ha fatto suonare un campanello? Stavo per dare al gelato il gusto di mela (apple), ma sarebbe apparso troppo ovvio. La spiacevole scena di cui sopra corrisponde al modo in cui io percepisco la mentalità di coloro che impongono il loro mondo agli altri, sotto l'apparenza di una "interfaccia utente standard".

Non mi preoccuperei così tanto se coloro che propongono una interfaccia utente standard limitassero la quantità di dettagli dell'implementazione. Essi non la pensano a questo modo. Iniziano discutendo se un file requester debba ordinare i file mentre carica, ordinare solo su richiesta o non ordinare del tutto. Vorrebbero dettarci il modo in cui i gadget devono apparire, se premere due volte il pulsante del mouse su un nome di file corrisponda a selezionare il gadget OK o LOAD (carica), se si debba selezionare la directory superiore con un gadget o come parte della lista dei file. Vorrebbero dettare il più piccolo dettaglio, sempre in nome di una interfaccia utente standard.

E' frustrante alle volte parlare a coloro che tengono vicino al proprio cuore l'interfaccia del Mac. Non è il fatto che amano l'idea di una interfaccia utente standard che mi urta, perché sono completamente d'accordo con l'idea che uno standard sia desiderabile, ma il fatto che provengono da un ambiente in cui è considerato accettabile che una persona o un gruppo di persone decidano che cosa debba essere lo standard, fino al dettaglio più insignificante.

Quei programmatori per i quali lo standard non è esattamente quello di cui hanno bisogno nel loro programma, sono naturalmente liberi di scrivere la propria interfaccia utente. Se il programma che ne risulta venderà o meno è tutta un'altra questione.
L'Apple, gli utenti e gli sviluppatori del Mac esercitano una buona dose di pressioni contro ogni modifica dell'interfaccia utente. Ciononostante, il programmatore ha certamente il diritto di far fare al suo programma ciò che gli sembra più conveniente.
Questa situazione implica molti problemi e anche se molti aspetti appaiono positivi, non vedo ragione perché non si possano trattenere i fattori positivi senza i problemi, per applicarli ad Amiga.

Un desiderio diffuso fra i possessori di Amiga è un file requester di sistema; così utilizzerò questo, come esempio di cosa debba essere fatto per promuovere uno standard, eliminando i problemi. Il beneficio primario di una interfaccia utente standard è che il nuovo utente può facilmente affrontare qualsiasi applicazione e comprendere immediatamente che essa funziona, in buona parte, come tutte le altre. Questo riduce la quota di FUD (Fear, timore, Uncertainity, incertezza e Doubt, dubbio) e permette, sia al novizio che al veterano, di acquisire velocemente familiarità con un nuovo programma.

Il problema si condensa, in verità, in una sola questione. Se a voi, come utenti della macchina, capita di non gradire l'interfaccia utente o una sua parte, è un vero peccato. Dovete dipendere da ciò che avete acquistato. Se un programmatore decide di scrivere un proprio file requester, e voi comprate il programma, anche da quello dovete dipendere. La sola differenza è che nel caso di un file requester fornito dal sistema, voi dovete dipendere dalla versione ufficiale e nel secondo dalla versione del programmatore.

Le prime domande che ci potremmo porre sono:
"Per chi è lo standard?"
"Possiamo immaginare un modo per avere uno standard e al tempo stesso delle possibilità di scelta?"
La risposta alla prima domanda, secondo me, è che lo standard è per l'utente. Un file requester built-in rende

sicuramente le cose più semplici anche al programmatore, ma in ultima analisi i benefici maggiori sono per l'utente.
La risposta alla seconda domanda è un sonoro "Sì!" ed è proprio questo che voglio chiarire con la proposta che segue. Si tenga in mente che io sono convinto che la CBM fornirà, in qualche futura release, un file requester built-in e che queste osservazioni intendono anche influenzare una sua possibile implementazione.

Questa proposta non riguarda il modo in cui l'utente interagisce con il file requester. Non voglio nemmeno occuparmi dell'aspetto del numero di gadget o della grandezza. La proposta riguarda unicamente l'interfaccia tra il programma chiamante e il file requester e tra un momento capirete perché.
L'idea generale è che un file requester fornito dal sistema dovrebbe essere configurabile in più modi, mediante l'uso di flag impostati dall'utente. Ogni programma, inoltre, dovrebbe avere la possibilità di chiamarlo con facilità, come qualsiasi altra routine di sistema. Dovrebbe fornire tutte le informazioni che uno potrebbe aspettarsi da un file requester, il che significa che dovrebbe fornire un path e un nome di file (o più nomi di file) al programma chiamante.

Il file requester dovrebbe anche essere facilmente sostituibile dall'utente finale. Si considerino le implicazioni di questo fatto. Un programmatore, se fornito di un file requester semplice da chiamare e dotato di funzionalità adeguate, deve trovarsi in una situazione molto difficile per avere delle ragioni che lo spingano a crearne uno proprio.
La sola ragione per cui i programmatori delle applicazioni per Mac e per Presentation Manager prendono in considerazione l'idea di farlo, deriva dal fatto di avvertire che il file requester fornito dal sistema è inadeguato, brutto o entrambe le cose.
Ogni inadeguatezza può essere superata stabilendo una convenzione di chiamata che sia completa e flessibile. Qualsiasi apetto sgradevole può essere superato permettendo di operare, con facilità, delle sostituzioni. La completezza è la parte più difficile di questo schema, quando si prenda in considerazione il fatto che la convenzione di chiamata dovrebbe consentire tutti i particolari parametri o flag che sembrano necessari, sia ora che in futuro.

Essendo i programmatori ciò che sono, senza dubbio qualcuno si farà avanti con un flag o un parametro che deve assolutamente essere presente, proprio quando le ROM sono in piena produzione e pronte per la spedizione. La seconda parte, la possibilità di rimpiazzare facilmente gli elementi del sistema è già sul posto, con font, keymap, printer driver, device driver, librerie e persino singole routine entro le librerie.
Vediamo ciò che occorre a un programmatore. Un'applicazione potrebbe aver bisogno di specificare che può accettare più di un nome di file o che può gestirne uno solo. Potrebbe aver bisogno di specificare: un titolo da far apparire sul requester, un path per la lista delle directory, se desidera che la lista sia filtrata per mostrare solo i file che rispettano un certo pattern (o solo quelli che non lo rispettano), e probabilmente altre cose che non mi vengono in mente in questo momento. Il programmatore ha bisogno di sapere se è stata fatta una selezione, se la selezione era multipla, se il requester è stato cancellato senza fare alcuna selezione e, naturalmente, il path e il nome del file.

Al programmatore non importa sapere quale aspetto assuma il requester, se la directory appaia ordinata o se sia stato usato come input, per effettuare la selezione, il mouse, la tastiera o la voce. Queste cose è meglio lasciarle all'utente. Fra le opzioni di configurazione, da impostare per esempio attraverso Preferences o un programma similare, si potrebbero includere cose come: possibilità di selezione immediata, selezione doppia o uso esclusivo del gadget "OK", posizione assoluta del requester sullo schermo o relativa al puntatore del mouse e così via. Fra le opzioni da impostare a questo modo dovrebbero esserci quelle che non riguardano il programmatore e costituiscono più una questione di gusto per l'utente. L'impostazione dovrebbe, in linea di principio, influenzare ogni file requester.

Siamo giunti al cuore della questione. Il programmatore ha a disposizione la sua chiamata di sistema e può fare affidamento sul risultato della sua chiamata. Se il file requester può essere sostituito facilmente, l'utente ha la possibilità di scegliere fra più file requester. Così, possiamo avere uno standard in più di un modo. Il primo standard è il file requester di sistema che i nuovi proprietari di Amiga vedranno. Sarà chiamato dalla maggior parte dei programmi, con l'eccezione dei programmi più vecchi che usano un proprio schema di funzionamento e quelli che non sono stati ancora aggiornati. Ciò fornirà un terreno comune per dialogare con il sistema e le applicazioni e impararne il funzionamento.
L'altro standard si ottiene quando un utente decide di sostituire il file requester fornito dal sistema con un altro di sua preferenza. Il sostituto, mentre appare per un verso come non standard, in realtà è, tutto sommato, uno standard, se viene chiamato ogniqualvolta si invoca il file requester di sistema.
Così tutte le applicazioni che chiamano il requester in maniera normale, chiameranno lo stesso file requester senza sapere o preoccuparsi del fatto che stanno chiamando un sostituto; con ciò tutte le applicazioni che lo useranno si comporteranno in maniera identica dal punto di vista dell'utente ed esattamente nel modo che l'utente preferisce.
Mi sembra che così si ottengano tutti i vantaggi di un file requester fisso, mentre si lascia ancora all'utente la possibilità di effettuare delle scelte.

Penso che le scelte siano importanti, in quanto sembra che ogni proprietario di Amiga abbia le sue preferenze a proposito di file requester. Personalmente, mi piacciono quelli che possono essere spostati sullo schermo e sono sufficientemente piccoli da permettere di vedere il testo che altrimenti sarebbe nascosto.

(segue a pag. 41)

Copiare i dischi

La tecnica sottile dell'uso del trackdisk.device

Bob Rakosky

Bob Rakosky è un consulente di Data Processing presso l'AGS Information Services Inc. ed è l'autore dell'utility saf-T-net (pubblicata da Metadigm) per il backup degli hard disk e della versione per Amiga del gioco Empire (pubblicato da InterStel).

Si può accedere ai floppy disk di Amiga a diversi livelli. Il modo standard per leggere da e scrivere su dischi è rappresentato dal FileSystem, che governa lo spazio su disco e mantiene traccia dei raggruppamenti logici dei dati chiamati file. E' attraverso il FileSystem che possiamo ottenere l'elenco dei file presenti in una directory, stabilire quanto spazio è attualmente libero sul disco, leggere i dati contenuti nei file e crearne di nuovi.

Il FileSystem è indipendente dal device (dispositivo). I dati, immagazzinati in qualsiasi device del FileSystem, appaiono come una serie di blocchi, ciascuno dei quali contiene 512 byte.
Tutto quello che c'è bisogno di sapere è il numero dei blocchi da 512 byte disponibili. I dati verranno letti e scritti come blocchi individuati da un numero, il cui valore parte da 0 e continua fino a n-1, dove n è il numero di blocchi contenuti nel device.

Dal momento che il mio desiderio è copiare un disco, potrei farlo usando il FileSystem, presupponendo d'avere un disco vuoto e formattato, sul quale scrivere tutti i file presenti nel disco sorgente.
Ma che cosa accadrebbe se non avessi un disco già formattato? E se il disco da copiare non fosse creato dal FileSystem dell'AmigaDOS, e non contenesse dati raggruppati in file? Anche se fosse un disco standard per il FileSystem, copiarlo file per file sarebbe piuttosto lento, specialmente se il disco contenesse un gran numero di piccoli file. Copiare a livello di device, piuttosto che attraverso il FileSystem, potrebbe essere molto più veloce finché copiamo da e verso device dello stesso tipo. Questo è il livello al quale intendo accedere ai floppy disk.

Esiste ancora un altro livello al quale è possibile accedere ai floppy disk: pilotare direttamente l'hardware. Siccome la mia intenzione è quella di accedere solo a dischi nel formato standard di Amiga, questo tipo di approccio oltrepassa le mie necessità e non sarà trattato dal presente articolo.

Organizzazione dei dati

Il driver trackdisk.device vede i dati come sono fisicamente contenuti nel mezzo meccanico, cioè come una serie di tracce di dati codificati.
Ciascuna traccia su un disco Amiga contiene 11 settori di dati (più un insieme extra di informazioni non usate dal FileSystem), e ciascun settore, dopo essere stato decodificato, contiene 512 byte di dati (si noti che non è necessaria un'esatta corrispondenza tra la lunghezza di 512 byte per blocco usata dal FileSystem e la lunghezza di 512 byte per settore usato dal trackdisk, ma ciò semplifica le cose).

L'attuale disposizione dei dati all'interno della traccia è irrilevante per i nostri scopi, in quanto di questo tipo di dettagli si prende cura il driver dei dischi.
Non è necessario conoscere come i dati vengano codificati, né come decodificare le informazioni contenute nel device fisico. Un determinato device viene configurato al fine di contenere un numero fisso di tracce per superficie, e un numero fisso di superfici. Per i floppy disk standard da 3.5" di Amiga, ciascuna superficie contiene 80 tracce e il disco contiene due superfici utilizzabili.
Per i floppy disk standard da 5.25" di Amiga, sono previste due superfici di 40 tracce ciascuna.

Le richieste di input e output vengono inviate al device trackdisk come indirizzi relativi di byte, che partono da 0 e continuano fino alla massima capacità del disco (880K per i dischi da 3.5"). Le richieste devono riguardare indirizzi che cadono sul confine di un settore (l'indirizzo, cioè, deve essere un multiplo di 512) e la lunghezza dei dati letti o scritti (sempre in byte) deve essere riferita ad interi settori (ancora un multiplo di 512).
Tuttavia, il driver del device leggerà o scriverà i dati solo come tracce complete; pertanto deve possedere un buffer interno in cui mantenere l'intera traccia.
Dal momento che noi leggeremo e scriveremo dischi interi, ha maggiore senso eseguire tutte le operazioni di I/O usando come unità la traccia completa, in quanto è questo il modo in cui i dati saranno trattati internamente dal driver.

Usare i driver dei device

Amiga permette il controllo dei suoi device periferici hard-

ware attraverso moduli software chiamati device driver. Il driver per i floppy disk è contenuto nel codice del Kickstart e vi si accede nello stesso modo in cui si accede a tutti i device di Amiga.
L'accesso al device viene realizzato utilizzando la funzione OpenDevice( ) della libreria Exec, alla quale va specificato il nome del device e il numero di unità. Dopo di che, i comandi di input e output verranno inviati al device usando le funzioni DoIO() e SendIO(), passando loro una struttura IORequest.
Queste due funzioni sono simili, tranne per il fatto che una chiamata a DoIO( ) non farà ritorno fino a che la corrente azione di input/output non sia stata completata, mentre SendIO( ) avvierà l'operazione di I/O, ma farà ritorno (se possibile) prima che sia completata.

Le comunicazioni tra il vostro task e il task (separato) che attua realmente il processo di I/O, vengono compiute attraverso le funzioni per il trasferimento dei messaggi fornite da Exec (la prima parte della struttura IORequest è una struttura Message di Exec) e nel caso di I/O asincrono, questo messaggio viene inviato dal task del device al vostro task attraverso la funzione ReplyMsg( ), utilizzando come MsgPort il valore mn_ReplyPort, contenuto nella struttura Message dell'IORequest. Si può controllare l'arrivo di questo messaggio attraverso le funzioni Wait( ) e GetMsg( ).

Una discussione completa dei device di I/O di Amiga occuperebbe molto più spazio di quello che questo articolo può permettersi. Una discussione completa, a livello generale, occupa un intero capitolo del ROM Kernel Manual. Ci limiteremo quindi a trattare solo le questioni direttamente connesse all'accesso dei floppy disk, come appaiono nel listato (sul disco) che accompagna questo articolo.

I/O con il trackdisk.device

Il trackdisk.device accetta tre gruppi di comandi: comandi standard per device, comandi specifici per trackdisk, comandi estesi specifici per trackdisk. Il programma esempio, copydisk.c, utilizza comandi che cadono nella prime due categorie. I comandi che discuteremo e useremo sono:

Comandi standard:

```
CMD_READ    --  Legge uno o più settori
CMD_WRITE   --  Scrive uno o più settori
```

Comandi specifici per trackdisk:

```
TD_MOTOR  --   Accende o spegne il motore
TD_FORMAT --   Inizializza una o più tracce
TD_GETNUMTRACKS -- Ritorna il numero totale di tracce
presenti sul device
```

Come abbiamo già affermato, l'I/O a livello di device viene eseguito attraverso l'uso di una struttura chiamata IORequest.

Il trackdisk.device usa una forma estesa di IORequest chiamata IOStdReq. Entrambe le strutture sono definite nel file include standard exec/io.h. Inoltre, i comandi estesi, specifici per trackdisk, usano una forma estesa della struttura IOStdReq. Questa struttura IORequest estesa, chiamata IOExtTD e definita nel file header devices/trackdisk.h, non viene usata nel programma esempio, perché i comandi che la richiedono non sono né necessari, né utilizzati nel programma. Si noti che la discussione che segue si riferisce alla struttura IORequest, senza fare distinzioni fra queste versioni estese.

Il programma Copydisk

Copydisk.c è stato scritto al fine di illustrare alcuni dei principi implicati dalla lettura o scrittura dei dischi a livello di device. Si tratta di un semplice programma per copiare i dischi, con qualche vantaggio in più rispetto al programma standard DiskCopy distribuito con il Workbench di Amiga. Il programma è stato progettato cercando di sfruttare la memoria disponibile nel sistema, per permettere di effettuare copie con un singolo floppy drive, limitando il più possibile il numero di cambi di disco. Inoltre, se il sistema ha abbastanza memoria per mantenere l'intero contenuto del disco sorgente, si possono effetuare copie multiple dello stesso disco senza essere costretti a rileggere il disco sorgente ogni volta.

Il programma deve essere lanciato solo da CLI. La linea di comando per eseguirlo ha la forma:

```
copydisk sorgente destinazione [MULTI] [NOVERIFY]
```

in cui sia la fonte che la destinazione sono validi identificatori di disk drive (come DF0: o DF2:).

L'opzione MULTI, se specificata, costringe il programma a non terminare dopo aver concluso la prima copia, ma a continuare richiedendo all'utente se devono essere effettuate copie multiple dello stesso disco. Questa modalità operativa cessa quando l'utente inserisce un CTRL-C da tastiera, come informa il prompt. Con l'opzione NOVERIFY i dati saranno scritti sul disco destinazione molto più velocemente, ma diminuirà il controllo sugli eventuali errori. Dovrebbe essere usato solo con cautela, perché è molto probabile che, in questo modo, errori in scrittura non vengano rilevati dal programma.

Ho deciso di usare alcune funzioni dell'ARP.library, la libreria, liberamente distribuibile, di funzioni di supporto creata da Charlie Heath, Scott Ballantyne e altri. Il programma usa il codice di startup della versione 1.1 di ARP (l'ultima versione disponibile nel momento in cui ho scritto il programma), invece del codice di startup standard distribuito con il pacchetto del compilatore. Il vantaggio offerto sta nel fatto che la libreria ARP viene aperta automaticamente dal modulo di startup, e che la linea di comando viene analizzata automaticamente, mediante la funzione Gads( ) di

ARP, prima che il controllo passi alla funzione main( ). Di conseguenza, l'elaborazione degli argomenti, nella funzione main( ), può con sicurezza presupporre che i puntatori agli argomenti nell'array argv[] stiano nell'ordine che corrisponde al template della linea di comando, specificato attraverso la variabile CLI_Template.

Il vantaggio principale derivato dall'uso della libreria ARP, è la riduzione della quantità di codice che ho dovuto scrivere. Il modulo di startup di ARP consente di avere un prompt e un'analisi della linea di comando molto sofistificati, senza dovere scrivere di propria mano il codice corrispondente.

Ho anche fatto uso della capacità di ARP di tenere una registrazione della memoria allocata (attraverso la funzione ArpAllocMem( )); ciò mi ha permesso di allocare una lista di buffer, senza doverne poi liberare la memoria, dal momento che viene liberata automaticamente durante la fase di conclusione del programma dal codice _exit contenuto nel modulo di startup (che si preoccupa anche di chiudere la libreria ARP). Ulteriori economie sono state realizzate usando le funzioni ARP per effettuare l'I/O a livello di packet del DOS, necessario per impedire al FileSystem di usare un disk drive (ci si riferisca a inhibit_drives( ) nel listato che accompagna l'articolo).

Tuttavia sono incorso in problemi potenziali usando il codice di inizializzazione di ARP. Nella funzione valid_devs( ) tento di esaminare la lista dei nodi di device del sistema, al fine di controllare la presenza del device specificato sulla linea di comando. Il punto di partenza di questa lista si trova attraverso la struttura RootNode, che è indirizzata (mediante un BPTR) dalla struttura DosLibrary. L'indirizzo di quest'ultima è costituito dal puntatore DosBase che viene ottenuto dal codice di startup con l'apertura della dos.library.

Tuttavia, quando si usa il codice di startup di ARP, DosBase contiene in realtà un puntatore alla base della libreria ARP (che contiene una copia parziale della libreria DOS). Sfortunatamente, questo "clone" non è una copia esatta e alcuni campi contenuti nella struttura DosLibrary non si ritrovano nella copia. Nella versione 1.1 di ARP, che io uso, il BPTR al RootNode non compare, sebbene io sia stato informato che nella versione 1.3 della libreria il puntatore al RootNode sarà disponibile attraverso questo pseudo-puntatore DosBase. Si noti che il "vero" puntatore DosBase è presente nella struttura della libreria ARP, e che ho bisogno di ricavarne il valore, per conoscere l'indirizzo di RootNode. Attraverso di esso, posso poi trovare la lista dei nodi di device del sistema e realizzare la convalida necessaria.

Esaminiamo le routine di I/O per i dischi

La prima cosa necessaria per accedere ai floppy disk è effettuare una chiamata a OpenDevice( ). Ciò inizializzerà una struttura IORequest con i valori necessari ad effettuare richieste di I/O. I valori impostati in tale struttura IORequest sono indispensabili per inviare richieste di I/O al driver: possono essere copiati in un'altra struttura IORequest, se necessario, (copiando l'intera struttura inizializzata da OpenDevice( )). Nel nostro caso, noi inizializzeremo una singola IORequest per il device di input e la useremo per tutti gli input. Lo stesso processo verrà seguito per il device di output.

Abbiamo bisogno di determinare il numero totale di tracce presenti nel device sorgente. Ciò ci permetterà di stabilire il numero massimo di cache buffer da allocare, dal momento che intendiamo caricare in memoria la massima quantità possibile di tracce del disco sorgente (per minimizzare il numero di cambi di disco necessari quando si copia da e verso lo stesso drive, e per permetterci di effettuare, se possibile, copie multiple con una sola lettura del disco sorgente).

Vogliamo anche verificare che il device di output utilizzi lo stesso numero di tracce, in quanto la logica del programma dipenderà da questo. Il comando TD_GETNUMTRACKS restituirà il numero totale di tracce disponibili sul disco in questione. Questo valore sarà disponibile nel campo io_Actual della struttura IORequest. Dopo aver ottenuto l'accesso al driver del device, dobbiamo informare il FileSystem dell'AmigaDOS di tenersi alla larga dai nostri dischi. Sarebbe pericoloso che il FileSystem accedesse al device quando è in corso una operazione di scrittura, dal momento che noi non ci preoccupiamo di garantire l'integrità logica dei file nel periodo che intercorre tra il momento in cui iniziamo a scrivere e quello in cui completiamo l'operazione di copia.

La richiesta "tieniti alla larga da" è effettuata dalla routine inhibit_drives(), che usa I/O a livello di packet del DOS per comunicare con il FileSystem. Siccome questo articolo riguarda principalmente l'accesso ai driver dei device, piuttosto che l'accesso agli handler del FileSystem, non abbonderò in dettagli a questo riguardo. Basterà dire che, inviando un packet ACTION_INHIBIT all'handler, con un argomento TRUE, il FileSystem considererà il disco uno standard "Not a DOS Disk" e non tenterà di accedere ad esso. L'icona del disco, se il Workbench è stato caricato, apparirà come "DFn:BUSY".

Dopo aver allocato i buffer necessari ad effettuare il cache dei dati, abbiamo bisogno di allocare ancora un buffer col quale leggere e scrivere i dati. Questo perché l'area dei dati da passare al trackdisk.device per le operazioni di input e output "deve essere in CHIP RAM". Ciò è richiesto dal trackdisk per il motivo primario che il driver del device usa il blitter di Amiga per effettuare la codifica e la decodifica dei dati (infatti i dati presenti fisicamente sul floppy disk non sono memorizzati come 8 bit per carattere, ma codificati usando una tecnica chiamata MFM, che aiuta ad assicurarsi che la serie dei bit sia interpretata nella maniera corretta). Noi non abbiamo allocato i cache buffer specificatamente in

memoria CHIP, perché è impossibile far entrare un intero floppy disk nell'area CHIP a disposizione (finché non sarà disponibile 1Mb di CHIP RAM).

Ciò ci conduce a un'altra decisione fatta in sede di progettazione del programma. Ho deciso di usare I/O sincrono, il che significa che le mie richieste di I/O non restituiranno il controllo finché non saranno state completate. Ciò viene realizzato usando la funzione DoIo(). Avrei potuto scegliere di effettuare I/O asincrono utilizzando la funzione SendIO( ), nel qual caso il mio programma avrebbe riottenuto il controllo dopo che la richiesta fosse stata accettata, ma prima del suo completamento. Ciò mi avrebbe permesso, per esempio, di leggere una traccia mentre ne sto scrivendo un'altra (assumendo che lettura e scrittura vengano operate su drive diversi). Avrei potuto anche inviare una seconda richiesta di lettura mentre la prima era ancora in via di elaborazione, ma ciò non avrebbe comportato un particolare aumento della velocità.

Sebbene il primo esempio (lettura e scrittura simultanee) avrebbe consentito un cospicuo miglioramento del programma, ho deciso di non implementarlo, in primo luogo perché avrebbe reso il programma molto più complicato, mentre volevo mantenere il codice semplice e lineare. Dovevo anche limitarne la lunghezza, per evitare che l'editore se la prendesse con me.
Come abbiamo già notato, il trackdisk effettua le operazioni di lettura e scrittura solo di multipli della lunghezza del settore. Tuttavia, siccome i dati vengono trasferiti da e verso il device solo come tracce complete, è più efficente realizzare il nostro input e output in maniera analoga e leggere e scrivere un'intera traccia alla volta (assumendo di posizionarci sempre a un offset corrispondente all'inizio della traccia).

La lettura dei dati è piuttosto semplice: prepariamo la richiesta usando il comando di I/O standard CMD_READ, specificando l'offset iniziale (espresso in byte) come il numero di traccia corrente (relativa a 0) per la lunghezza della traccia, e richiediamo di leggere la lunghezza di una traccia. Questa sarà letta nel nostro buffer di memoria CHIP e poi copiata nel cache buffer.
La scrittura di una traccia avrebbe potuto essere effettuata allo stesso modo, usando il comando standard CMD_WRITE. Il lettore avrà forse osservato, tuttavia, che il programma d'esempio non usa questa tecnica. Il programma copydisk usa invece il comando TD_FORMAT, specifico per il trackdisk.device.

Forse è necessaria qualche spiegazione. Sicccome è sempre possibile effettuare un CMD_WRITE per un singolo settore di dati, il trackdisk implementa le operazioni di scrittura in una maniera un po' diversa da quanto ci si potrebbe aspettare. Al fine di scrivere qualsiasi dato su una traccia del disco, il trackdisk dapprima legge l'intera traccia dal disco e poi rimpiazza nel suo buffer interno il settore (o i settori) che il programma intende attualmente scrivere. Infine trasferisce l'intera traccia dal buffer al disco. Nel nostro caso, leggerebbe l'intera traccia, rimpiazzerebbe l'intera traccia nel suo buffer con i dati che intendiamo scrivervi (dal momento che stiamo scrivendo tracce complete) e poi invierebbe i dati in output.

Questo funzionerebbe se il disco su cui stiamo scrivendo è già formattato, altrimenti la lettura iniziale fallirebbe e la scrittura non sarebbe eseguita. Inoltre, sarebbe notevolmente più lento, perché richiederebbe la lettura di un'intera traccia prima della scrittura. Esiste un'alternativa, tuttavia, dal momento che noi effettuiamo scritture di tracce intere. Il comando TD_FORMAT scrive semplicemente un'intera traccia di dati, sovrascrivendo qualsiasi cosa si trovi sul disco. Ciò funzionerebbe anche su un disco non formattato, dal momento che è questo il comando che permette di formattare i dischi. La sola restrizione è che l'operazione deve essere eseguita traccia per traccia.

L'aspetto negativo dell'uso di TD_FORMAT sta nel fatto che il comando sembra avere successo anche quando esiste un problema con il disco (come un disco corrotto). Questo sarà solamente rilevato da una successiva lettura della traccia. Ecco il motivo per cui il programma invia una richiesta di CMD_READ dopo TD_FORMAT, a meno che l'utente non decida di evitarlo specificando l'opzione NOVERIFY sulla linea di comando. Si noti che, con la verifica, le operazioni non sono più veloci di quelle realizzate con CMD_WRITE; tuttavia noi riusciremo a fare un controllo migliore dei dati presenti sul disco (dal momento che li leggiamo "dopo" averli scritti). Questa tecnica, inoltre, funziona anche con dischi non formattati, dove quella che fa uso di CMD_WRITE fallirebbe inevitabilmente.

Un altro comando richiede qualche commento: TD_MOTOR. Come abbiamo già notato, il comando serve ad accendere o spegnere il motore. Il lettore avrà forse notato che il listato non contiene alcuna richiesta di I/O per accendere il motore. Ciò avviene perché il trackdisk.device accende automaticamente il motore ogni volta che è necessario. Tuttavia, non lo spegne, a meno che il programma applicativo non lo richieda esplicitamente. Di conseguenza, abbiamo dovuto includere richieste per spegnere il motore nei momenti appropriati, come quando desideriamo che l'utente sostituisca il disco presente nel drive. Non è un'idea particolarmente buona quella di lasciare il motore acceso (e il LED corrispondente acceso) e chiedere all'utente di cambiare i dischi. Si noti che non spegniamo il motore dopo ogni operazione di lettura o scrittura, in quanto ciò rallenterebbe notevolmente le operazioni. Il motore impiega una significativa quantità di tempo a raggiungere la piena velocità e vogliamo incorrere in questo inconveniente solo quando è realmente necessario. Infine, si sarebbe dovuto notare che, facendo una copia del disco, traccia per traccia, il disco risultante è un esatto duplicato dell'originale. Il FileSystem di Amiga tiene traccia dei dischi che può vedere e si confonde se due dischi risultano identici.
Ciò generalmente produrrà un Guru o un crash di sistema.

Per evitarlo basta assicurarsi che il FileSystem veda qualche differenza tra i due dischi. Pertanto, ci assicuriamo che la data di creazione del disco (contenuta nel blocco radice) sia diversa nei due dischi. Tutto ciò può vedersi nel codice contenuto nelle funzioni write_track( ) e touch_root_blk( ) nel programma di esempio.

La doppia personalità del codice sorgente

Copydisk è stato sviluppato utilizzando la versione 5.02 del Lattice C, nel modo standard con interi a 32 bit. Tuttavia è stato intenzionalmente scritto per permettere che sia compilato usando la versione 3.6a del compilatore Manx, nel suo modo standard a 16 bit. A causa di questa decisione a livello di progettazione, certi costrutti possono apparire un po' più complicati di quelli cui si è abituati. Per esempio, ho deciso di fare uso della capacità di function-prototyping del compilatore Lattice, non presente attualmente sull'AZTEC C. Di conseguenza, ho usato la macro __ARGS( ) per dichiarare i prototipi. Questa macro è definita nell'header file stdio.h del pacchetto Lattice, nel seguente modo:

```
#define __ARGS(a) a
```

ed è (da me) definita per il compilatore Manx (entro un blocco #ifdef AZTEC_C) come:

```
#define __ARGS(a) ()
```

Questo permette di definire i prototipi delle funzioni nella forma:

```
int read_track __ARGS((struct cache *));
```

che sarà espansa, nel Lattice, come:

```
int read_track (struct cache *);
```

e, nel Manx, come:

```
int read_track ();
```

che sono forme che i rispettivi compilatori accetteranno come definizioni di funzioni.

La differenza nella lunghezza degli interi nei modi di default dei due compilatori genera alcune incompatibilità. Questo si traduce in cast aggiuntivi che potrebbero non apparire necessari in una implementazione o nell'altra, ma permettono al codice sorgente di funzionare correttamente su entrambi i sistemi.

In conclusione

Copydisk.c illustra alcune tecniche usate per accedere ai drive dei floppy disk a livello di device. E' anche un utile programma per copiare i dischi, sebbene questo sia un motivo secondario per giustificarne l'esistenza. A causa di ciò vi sono alcuni difetti nel programma così com'è. Vi sono anche più opportunità (lasciate al lettore come esercizio) per migliorare sostanzialmente il programma e le sue prestazioni.

E' necessaria una migliore logica di gestione degli errori, mentre fra le migliorie desiderabili sta l'aggiunta di una logica di I/O asincrona (che non è così banale come si potrebbe pensare). Inoltre, sulla mia lista dei desideri compare anche una migliore interfaccia, che dovrebbe rispettare lo stile Intuition in modo che il programma non sia limitato all'uso da CLI. Ma è solo un inizio!

(segue da pag. 36)

ViewPort

Altri preferiscono quelli grandi capaci di visualizzare un grande numero di nomi di file. Alcuni vogliono usare la tastiera o il mouse o entrambi. Preferite i gadget per la selezione del nome del volume o una lista? Volete che le assegnazioni di directory appaiano assieme ai nomi di volume?

Tutte queste scelte non riguardano il programmatore, che può stare sicuro che quando richiede il nome di un file, l'utente glielo fornirà nel modo che appare più comodo per l'utente, non alla Commodore o all'applicazione.

Non è necessario limitare la sostituzione di alcun particolare visivo; in verità non è affatto necessario che appaia sullo schermo. Una persona debole di vista potrebbe desiderare un file requester ampio, anche se solo tre o quattro file possono essere visti contemporaneamente. Un cieco potrebbe volere un file requester sonoro privo di qualsiasi rappresentazione visiva. Di fatto, quasi ogni concepibile metodo per la presentazione di liste di file potrebbe essere gestito da un sostituto.

Un terminale collegato alla porta seriale potrebbe fare buon uso di un file requester tradizionale con menu numerico, che potrebbe essere selezionato attraverso sequenze di caratteri senza dover correre attraverso la stanza o attraverso la città per effettuare la selezione alla tastiera di Amiga.

Si ricordi che, sebbene io stia parlando di un file requester, il principio potrebbe essere applicato ad altri aspetti dell'interfaccia utente. Penso che possiamo avere degli standard e permettere ancora la creatività. Possiamo essere facilitati nell'apprendimento di una macchina che può cambiare man mano che l'utente avanza. Possiamo avere una macchina senza paragoni per quanto riguarda la sua capacità di accettare un'ampia gamma di preferenze e possibilità.

E così, chi ha qualche grande idea per un file requester?

Linguaggio Assembly

Parte 3 - leggere un file e alcune riflessioni sulla purezza

Jim Butterfield

Jim Butterfield non ha bisogno di presentazioni. Il suo nome costituisce un punto di riferimento per gli utenti Commodore di tutto il mondo.
Il suo interesse per i computer risale al KIM-1 da 1K. L'enciclopedica competenza di Jim sui prodotti Commodore è testimoniata dai suoi articoli, libri, lezioni e persino programmi televisivi.

"Caro Jim,
nella prima parte, hai implicitamente affermato che un'istruzione MOVE da un registro dati a un registro degli indirizzi possa essere seguita da un salto che dipende dal dato mosso. Non è così."

"Caro Lettore, mi correggo: qualsiasi istruzione che faccia riferimento a un registro degli indirizzi come operando destinazione, non influisce sui flag che vengono testati da un comando di salto (Brach).
Va notato che il salto cui Lei si riferisce (quello che segue l'apertura della libreria DOS) [si veda Transactor per Amiga, n.5, p.25, N.d.T.] funziona correttamente sull'attuale configurazione del sistema. Accade, infatti, che i flag siano correttamente impostati dalla stessa routine OpenLibrary. Si provi a definire "dos.library" in maniera errata, e si potrà vedere il salto che permette di evitare l'esecuzione delle restanti linee del programma. Ciononostante, non si tratta di un codice del tutto rigoroso. Cercherò di stare più attento in futuro."

File: preliminari

Questa volta leggeremo un file e poi invieremo il suo contenuto verso lo schermo. Prima di iniziare, dobbiamo però affrontare il concetto di "memoria di lavoro". I nostri primi programmi esemplificativi evitavano questa questione: è venuto il momento di immergersi in essa.

Se avete programmato su computer "più semplici", potreste aver usato il semplice metodo di definire l'area di lavoro di cui avevate bisogno, per poi usarla. Ciò può funzionare anche su Amiga, ma vi sono molte altre possibili soluzioni che potrebbero realizzare meglio l'obiettivo.
Abbiamo bisogno di approfondire i metodi di programmazione, ma anche di chiarire l'architettura dei programmi di Amiga. E' un itinerario che vale la pena di percorrere.

Hunk e struttura dei programmi

Sui computer più semplici un programma è posto in un'area di memoria la cui posizione è generalmente conosciuta in anticipo; ma la regola più importante è che l'intero programma è collocato in un unico posto. Amiga non potrebbe avere un multitasking effettivo se ciascuno programma insistesse per essere caricato a un particolare indirizzo in memoria. Non ci sarebbe alcun modo per risolvere i conflitti tra le richieste dei diversi programmi. Così, come voi senza dubbio già sapete, i programmi per Amiga sono in grado di essere rilocati al momento del caricamento.

Inoltre, su Amiga, i programmi non hanno bisogno di essere caricati come un singolo segmento di memoria. Un singolo programma su disco può essere costituito da molti "hunk". Quando il programma viene caricato, gli hunk vengono separati; ciascuno di essi occuperà un diverso segmento di memoria. Il loader di Amiga garantirà che tutti i pezzi siano connessi fra loro. Perché usare gli hunk, invece di una singola area di memoria per il programma? Ci sono diverse ragioni. In primo luogo, si risparmia ad Amiga la preoccupazione di trovare un pezzo di memoria abbastanza grande da contenere l'intero programma: i pezzi di memoria possono essere trovati dovunque siano. In secondo luogo, certi dati potrebbero richiedere tipi particolari di memoria: per esempio, dati destinati al video, all'interno del vostro programma, potrebbero avere bisogno di situarsi nella memoria CHIP, anche se la parte restante del programma può essere posta in qualsiasi tipo di memoria. Infine, cosa forse più importante, il sistema degli hunk pone le basi per lo sviluppo futuro dell'architettura di Amiga.

Come si ottengono gli hunk? Potete costringere l'assembler a spezzare il vostro codice in più hunk, se lo desiderate. Normalmente ciò viene fatto se si desidera sviluppare e assemblare il codice in pezzi separati, per poi chiedere al linker di connetterli tra loro.
Rimane comunque possibile, con la maggior parte dei linker, fondere molti piccoli hunk in un singolo hunk.

Storia degli hunk, tipi di hunk

Per ragioni, soprattutto, di tipo storico, vi sono tre tipi di hunk: code, data e bss. Quest'ultimo termine deriva dall'e-

spressione, quasi priva di significato, "Base of Stack Segment" (base del segmento dello stack); molti anni fa, lavorai su un sistema che usava bss con il significato di "Block Storage Size" (lunghezza del blocco da riservare).
L'idea era la seguente: le istruzioni del programma vanno nell'hunk code; i dati predefiniti e "fissi" vanno nell'hunk data; e sebbene l'hunk bss non contenga alcun valore (solo la definizione dell'ampiezza della memoria necessaria), esso permetterà di stabilire un'area di memoria separata per la vostra area di lavoro. Seguiamone la logica ancora più a fondo: non ci si aspetta che il vostro programma modifichi il contenuto degli hunk code o data; tutti i valori da modificare vanno nell'area bss. Potreste disassemblare un hunk code e non trovarvi che istruzioni; potreste effettuare un dump esadecimale di un hunk data e vedere solo dati, non codice.

Queste regole non vengono mai seguite con precisione assoluta. Alcuni compilatori producono obbligatoriamente i tre tipi di hunk, dopo di che il linker potrebbe fondere alcuni di essi o anche eliminare gli hunk vuoti. Non è necessario che i dati vadano in hunk di dati. Se si mettono i dati nell'hunk di codice, specialmente stringhe come "dos.library", si può accedere alle informazioni per mezzo dell'indirizzamento relativo al PC, risparmiando memoria e tempo.
La memoria di lavoro, che corrisponde all'area bss potrebbe essere ricavata da qualsiasi tipo di hunk. Ma, come vedremo, si tende a ricavare la memoria di lavoro al di fuori del programma stesso.

Le regole per l'uso dei tre tipi di hunk sono quasi scomparse. Per i piccoli programmi, in particolare, tutto potrebbe essere collocato in un singolo hunk code: programma, dati e memoria di lavoro. Il nostro esempio farà proprio così... ma noi lo useremo per studiare più da vicino la questione della memoria di lavoro.

Memoria di lavoro e purezza

La maggior parte dei programmi hanno bisogno di un posto per riporvi i dati. Questo potrebbe contenere cose come variabili, puntatori o buffer di dati. Ci sono almeno quattro modi per ricavare memoria di lavoro:

1) usare un hunk bss per ricavare un'area di memoria;

2) definire dello spazio entro il proprio programma, in una sezione di codice o di dati;

3) chiedere spazio allo stack di sistema (utilizzando LINK e UNLK);

4) chiedere al sistema di allocare un blocco di memoria (utilizzando AllocMem e FreeMem).

Con i primi due metodi lo spazio si trova all'interno del nostro programma. Gli ultimi due implicano dello spazio esterno: lo richiediamo, ed esso ci viene concesso dal sistema.

Ha qualche importanza il metodo usato? Si, se vogliamo che il codice sia "puro".
Il codice puro potrà eseguire più compiti contemporaneamente; se desideriamo rendere il nostro programma "residente", il codice deve essere puro. Ed ecco l'obiettivo: scrivere un programma in grado di fare il dump di un file verso lo schermo.
Per far ciò il programma ha bisogno di caricare dati da un file in un buffer. Andiamo ancora più in là: si supponga di volere che il programma sia in grado di lavorare su due file contemporaneamente. Con un singolo buffer, nasce un problema: un file si sovrapporrà all'altro, distruggendo il contenuto del buffer e il lavoro sui due file non sarà mantenuto distinto.

Il compito del programma

Il nostro programma sara chiamato DUMP. Quando l'utente imposta da CLI, "DUMP FILENAME", il contenuto del file sarà stampato sullo schermo. I caratteri non visualizzabili saranno trasformati in punti, tranne NewLine, che si comporterà nel solito modo.

Una cosa ancora: dal momento che il file potrebbe essere molto lungo e noioso, deve essere possibile bloccare il dump del file. Se viene rilevata la combinazione di tasti CTRL-C, l'esecuzione deve terminare.
La prima versione di questo programma "non" sarà pura. Mettiamoci al lavoro.

Codice di inizializzazione

Inseriamo linee di commento con data e nome, in modo che possiamo ricordarci qualcosa sul programma, quando ritorneremo ad esaminarlo in futuro.
I nostri dati di identificazione saranno molto brevi, per risparmiare spazio sulla rivista. Saranno seguiti dalle definizioni XREF; la maggior parte l'avete già vista in precedenza.

```
; Programma Dump - Jim Butterfield.  Marzo 15/89.
; Funzioni della libreria Exec
  xref _LVOOpenLibrary     ;-$228
  xref _LVOCloseLibrary    ;-$19E
  xref _LVOSetSignal       ;-$132
; Funzioni della libreria DOS
  xref _LVOOpen            ;-$1E
  xref _LVOClose           ;-$24
  xref _LVOOutput          ;-$3C
  xref _LVOWrite           ;-$30
  xref _LVORead            ;-$2A
```

Si tenga presente che possediamo una linea di comando, la cui lunghezza si trova nel registro d0 e l'indirizzo è in A0. Dobbiamo salvare queste informazioni... per il momento, le

copiamo in A4 e D4.

```
  ;-- Inizializzazione:
Startup
  move.l a0,a4 ; Puntatore alla linea di comando
  move.l d0,d4 ; Lunghezza
```

Gli assemblatori svolgono un mucchio di "riflessioni" al nostro posto. La prima istruzione MOVE trasferisce informazioni a un registro degli indirizzi.
Formalmente parlando, dovrebbe essere un'istruzione MOVEA. Ma il compilatore riesce a capirlo e trasforma l'istruzione per noi. Ora apriamo la libreria DOS nel solito modo.

```
lea     dosname(pc),a1 ;nome 'dos.library'.
clr.l   d0             ; Qualsiasi versione (0)
move.l  $4,a6          ; Usiamo la libreria Exec
jsr     _LVOOpenLibrary(a6); Apriamo la libreria DOS
move.l  d0,a6          ; Salviamo il puntatore al DOS
tst.l   d0             ; Controlliamone il valore
beq.s   StartupQuit    ; Usciamo in caso di errore
```

Se noi non facciamo quest'ultimo salto, significa che il DOS è stato aperto correttamente e che possiamo continuare con la fase seguente.
Questa sarà eseguita come una subroutine, chiamata attraverso BSR (Branch SubRoutine, chiama la subroutine). Sappiamo che la routine si trova molto vicino, così comunichiamo all'assembler che ci aspettiamo un salto corto (.s).

```
bsr.s  DOSinit
```

Quando ritorniamo dalla subroutine, il lavoro è già stato svolto. Il programma chiude la libreria DOS e termina.

```
  move.l  a6,al    ; Puntatore al DOS in al
  move.l  $4,a6    ; Usiamo la libreria Exec
  jsr     _LVOCloseLibrary(a6) ; Chiudiamo il DOS
StartupQuit
  rts              ; Fine del programma
```

Ed ora la nostra subroutine di primo livello. Non svolge ancora il compito principale. Ricaviamo l'handle per l'output, che ci permetterà di scrivere i risultati ottenuti.
Poi mettiamo un carattere di zero binario alla fine della linea di comando, che, probabilmente, ha un carattere di NewLine in quella posizione. Abbiamo bisogno di uno zero binario per creare una stringa del tipo C; è quello che il DOS pretende per il nome del file che dobbiamo passargli.

```
  ;-- Ricaviamo l'handle per l'output relativo al CLI:
DOSinit
  jsr      _LVOOutput(a6) ; Ricaviamo l'handle di
                            output
  move.l   d0,a5      ; e salviamolo
  ;
  ;-- Trasformiamo il nome del file della linea di
     comando in una stringa C:
checklen
  move.b  #0,-$1(a4,d4.W) ; Poniamo uno 0 binario in
                            fine
```

L'ultima istruzione si riferisce a un indirizzo ottenuto mediante la combinazione di più elementi. Comincia con A4, l'indirizzo della linea di comando.
Gli aggiunge D4, la lunghezza della linea di comando: possiamo usare ".W", lunghezza word, dal momento che possiamo essere sicuri che la linea di comando non contiene migliaia di caratteri! La somma di questi due registri ci porterà oltre la posizione in cui intendiamo andare, così torniamo indietro con uno scostamento di -1.

Non ci dovrebbe essere normalmente alcuno spazio all'inizio della nostra linea di comando, ma noi la controlleremo ugualmente per rimuoverli se è il caso.
Useremo il modo di indirizzamento "indiretto con incremento successivo".
Indiretto significa che usiamo il contenuto di A4 come un indirizzo; con incremento successivo significa che verrà aggiunto 1 all'indirizzo dopo averlo usato.
Quando abbiamo finito torniamo indietro di una posizione.

```
skipspc
  cmp.b   #$20,(a4)+
  beq.s   skipspc
  subq.l  #1,a4
```

Ora siamo pronti ad aprire il file. Il manuale di riferimento mostra il comando Open come:

```
File = Open( Nome, ModoDiAccesso)
D0           D1    D2
```

In altre parole, forniamo un puntatore al nome del file nel registro D1, un "modo di accesso" in D2 e poi chiamiamo la funzione DOS Open. Otterremo come risultato, un filehandle nel registro D0.
Se possedete un assembler con i file include, i valori corrispondenti al modo di accesso sono definiti nel file dos.i.
Per il momento, tutto ciò che dobbiamo conoscere è che il valore decimale 1005 (MODE_OLDFILE) specifica che intendiamo aprire il file in lettura.

```
move.l a4,dl      ; Puntatore al nome del file
move.l #1005,d2   ; MODE_OLDFILE (per la lettura)
jsr    _LVOOpen(A6)
move.l d0,d6      ; Handle del file di input
beq.s  DOSquit    ; Errore, si termina
```

Se il salto non viene effettuato, il file è aperto e possiamo metterci al lavoro.
Ora chiamamo la routine che svolge la parte più importante del lavoro con un'altra BSR (Branch SubRoutine, chiama la subroutine).

```
bsr.s DumpFile  ; La sezione principale
```

Quando saremo ritornati, dovremo rimettere tutto a posto. Ciò significa che dobbiamo chiudere il file. Close richiede solo il filehandle nel registro D1.

```
   move.l    d6,d1          ; Usiamo il filehandle...
   jsr       _LVOClose(A6)  ; per chiudere il file
DOSquit
   rts                      ; Ritorniamo.
```

Siamo finalmente sulla giusta strada. Il DOS è attivo (A6 contiene l'indirizzo della libreria), il file è aperto (il suo filehandle è in A5) e noi siamo pronti a leggere i dati dopo un po' di contabilità.

Non vogliamo che il testo sia stampato oltre il margine destro dello schermo. Contiamo i caratteri nel registro D7 in modo che possiamo inviare un carattere di NewLine se il numero diventasse eccessivamente alto.

```
      ; Il DOS è aperto e abbiamo gli handle di input/
      output.
      ; Partiamo!
   DumpFile
      moveq   #0,d7 ; Azzeriamo il contatore delle
      colonne
```

Ora leggiamo un blocco di dati dal file. Il comando Read è definito come segue:

```
LunghezzaAttuale =  Read(File,Buffer,Lunghezza)
   D0                     D1    D2     D3
```

Così mettiamo il filehandle nel registro D1, l'indirizzo del nostro buffer di dati nel registro D2, la lunghezza del buffer in D3 e chiamiamo la funzione Read del DOS. Al suo ritorno, avremo in D0 il numero di byte letti. Se il valore è inferiore a D3, siamo giunti all'ultimo blocco.

Nella prossima sezione di codice ci riferiremo all'indirizzo del buffer molte volte, come pure all'indirizzo del buffer di output. Per risparmiare codice, terremo quegli indirizzi rispettivamente nel registro A4 e A3. Leggiamo un po' di dati.

```
ReadBlock
   move.l d6,d1          ; Filehandle di input
   lea    BufAdr(pc),a4; Buffer di input
   lea    OutBuf(pc),a3
   move.l a4,d2          ; Copiamolo in D2 per Read
   moveq  #BufSize,d3    ; Lunghezza del buffer di input
   jsr    _LVORead(A6)   ; Leggiamo il file
   move.l d0,d5          ; Lunghezza dei dati letti
```

Abbiamo i dati. Sono nel buffer (il cui indirizzo si trova in A4). E' tempo di analizzarli, contarli e mandarli in output. Il registro D4 percorrerà l'area del buffer, non dobbiamo permettergli di superare il valore contenuto in D5, che costituisce la fine del buffer. Dopo il confronto, l'istruzione BCC (Branch Carry Clear, salta se il carry è a zero) funziona come "salta se maggiore o uguale a (senza segno)". Se conoscete il 6502, potrebbe sembrarvi un controsenso: il flag di carry è trattato in maniera diversa nel mondo del 68000.

```
   ; Qui operiamo l'analisi dei caratteri e l'output
   moveq   #0,d4  ; Puntatore all'inizio del buffer
   ; Loop dei caratteri - siamo alla fine del buffer?
NextChar
   cmp.l   d5,d4
   bcc.s   EndBuf
   ; Abbiamo i dati! Elaboriamoli.
   move.b  $0(a4,d4),d0
```

Il byte più basso del registro D0 ora contiene il carattere del buffer. Aggiorniamo il puntatore in D4 e cominciamo ad esaminare il carattere: è un carattere di NewLine? Non è stampabile?

```
addq.l  #1,d4
cmp.b   #$0a,d0
beq.s   NuLiner
```

Se troviamo il carattere 10 (0A esadecimale), si tratta di un carattere di NewLine: possiamo dunque saltare oltre e azzerare il nostro contatore dei caratteri per riga. Si noti che l'istruzione di confronto (CMP) ha il formato byte. Di nuovo, per essere formalmente corretto il comando dovrebbe essere essere CMPI, ma l'assembler si prenderà cura di questo al nostro posto.

```
cmp.b   #$1F,d0
bgt.s   NotDot
```

Confrontiamo il carattere con il numero 31, 1F esadecimale. Si noti che BGT (Branch Greater Than, salta se maggiore di) salterà se D0 è piu grande di 31. E' un confronto con segno, e dobbiamo parlarne un momento.

Numeri con segno

Quando il contatore del vostro registratore o videoregistratore mostra 9999, voi lo intendete (solitamente) come un valore di -1. Il concetto è il seguente: i numeri molto alti possono essere considerati come numeri negativi se si seguono delle regole particolari.
Se chiediamo al computer di trattare un valore come dotato di segno (signed), di contro a privo di segno (unsigned), facciamo esattamente questo. La regola funziona così: se il bit più alto del numero in questione è "impostato", l'intero numero verrà cosiderato negativo. Se decidiamo di usare questa regola, il valore contenuto in un byte non varierà tra 0 e 255, ma tra -128 e +127. Il computer tratta tutti i numeri allo stesso modo, ma i confronti avverranno in maniera

diversa nel caso di quantità con segno.

Si consideri la nostra situazione: desideriamo rigettare ogni carattere con un valore ASCII inferiore a 32 decimale. Sono quelli i caratteri non stampabili, naturalmente. Si consideri: il normale set dei caratteri ASCII si ferma a 127. A che cosa corrispondono quelli con valori più alti? Alcuni sono caratteri di controllo, e altri costituiscono il set di caratteri "alternativo". Un programma come DUMP non ha bisogno di stampare questo tipo di caratteri. Potremmo eliminare questi caratteri in più (mutandoli in punti) con un secondo test; ma abbiamo una scelta migliore. Se noi consideriamo i caratteri come se fossero dotati di segno, tutti questi 128 caratteri diventano negativi. Un confronto con segno li individuerà immediatamente.

Di fatto non è il confronto che è dotato di segno; è il modo che noi seguiamo per fare il test dopo il confronto. E BGE fa proprio al caso nostro; i caratteri speciali al di sopra del normale set ASCII non saranno stampati, ma trasformati in punti.
Ci vuole del tempo perché divengano familiari i diversi modi in cui i confronti vengono risolti. Se non abbiamo un carattere pari a 32 o più alto, lo trasformeremo in un punto (2E esadecimale, 46 decimale).

```
move.b  #$2e,d0
```

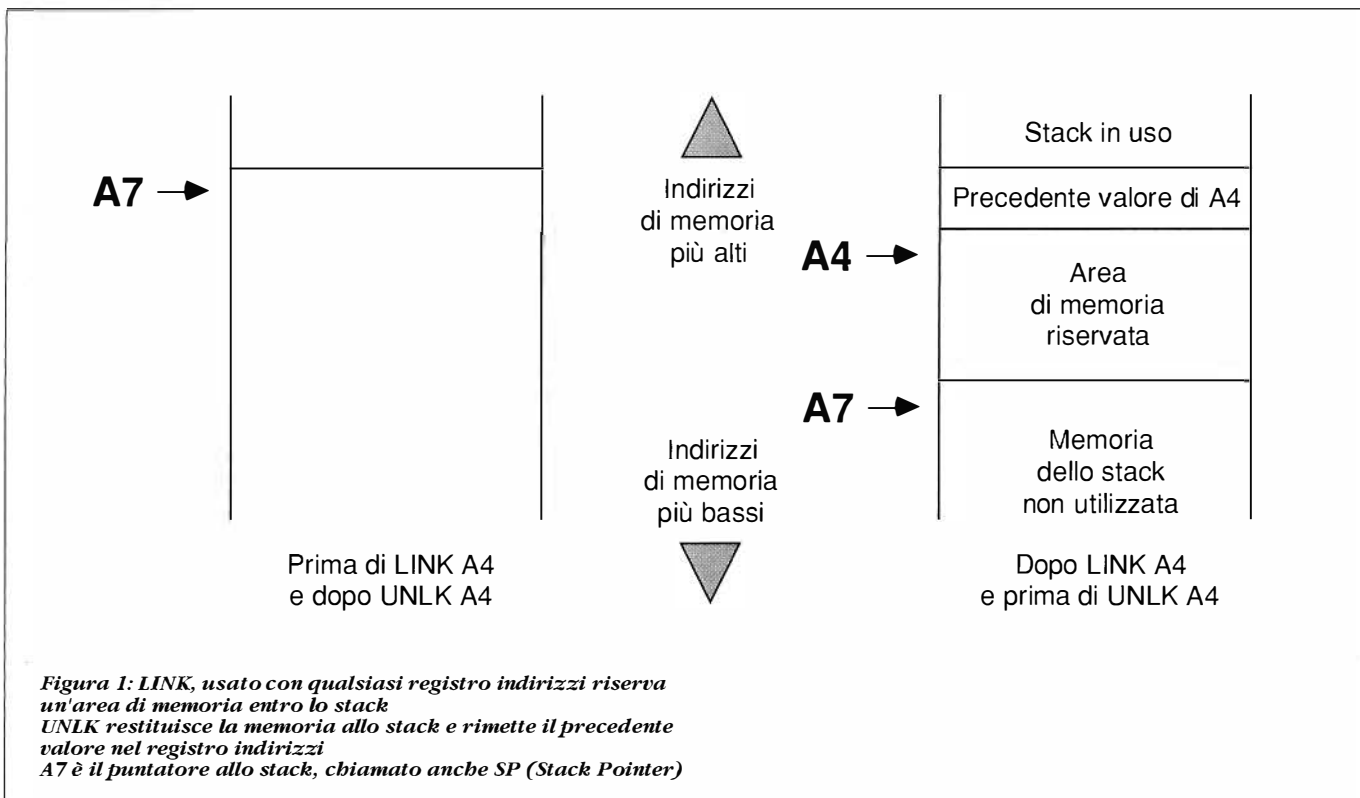
Che sia o non sia stato modificato, stamperemo il carattere (mediante Write) e incrementeremo il contatore dei caratteri di uno. Si ricordi che abbiamo posto l'indirizzo del piccolo buffer di output nel registro A3, cosa che ci risparmia del lavoro in questa situazione.

```
NotDot
  move.b  d0,(a3)
  move.l  a5,d1         ; Filehandle
  move.l  a3,d2         ; Buffer di output
  moveq   #1,d3         ; Lunghezza
  jsr     _LVOWrite(A6)
; Calcoliamo e inviamo NewLine se necessario
  addq.b  #1,d7         ; Aggiorniamo il contatore
  cmp.b   #64,d7        ; Siamo al limite?
  bls.s   NextChar
```

Se il numero dei caratteri ha superato 64 (un valore scelto senza nessuna particolare ragione), il salto qui sopra non avverrà (BLS sta per Branch Less Than, salta se minore di). Invece, eseguiremo NuLiner e ricominceremo con una nuova linea. Giungeremo qui anche nel caso incontrassimo un carattere di NewLine all'interno del file.

```
NuLiner
  move.b  #$0a,(a3)
  move.l  a5,d1         ; Filehandle
  move.l  a3,d2         ; Buffer
  moveq   #1,d3         ; Lunghezza
  jsr     _LVOWrite(A6)
```

Fra poco, torneremo indietro per leggere il prossimo carat-

Figura 1: LINK, usato con qualsiasi registro indirizzi riserva un'area di memoria entro lo stack
UNLK restituisce la memoria allo stack e rimette il precedente valore nel registro indirizzi
A7 è il puntatore allo stack, chiamato anche SP (Stack Pointer)

tere e continuare il dump del file. Ma prima, controlliamo se non sia giunto un CTRL-C.
Il test viene effettuato chiamando una funzione Exec, SetSignal. Non è documentata molto bene dal manuale ROM Kernel, tuttavia il codice che segue realizza l'obiettivo che ci proponiamo.
In primo luogo, si ricordi che stiamo per chiamare la libreria Exec e, dunque, abbiamo bisogno di un puntatore a ExecBase nel registro A6. Faremo bene a salvare il puntatore a DOSBase, ponendolo nel registro D7.

```
move.l  a6,d7                   ; Salviamo DOSBase
moveq   #0,d0
move.l  #$1000,d1
movea.l $4,a6
jsr     _LVOSetSignal(A6)       ; Testiamo CTRL-C
move.l  d7,a6                   ; Recuperiamo DOSBase
```

Abbiamo posto la maschera per CTRL-C in D1 e chiamato SetSignal, poi abbiamo recuperato il puntatore a DOSBase nel modo più veloce possibile.

Ora esaminiamo il risultato in D0. Ne estraiamo un singolo bit usando l'operazione AND, si pensi ad essa come a "rimuovi tutti i bit, tranne quelli cui sono interessato".

```
andi.l  #$1000,d0
tst.l   d0
beq.s   NotCtrlC
```

Se D0 contiene 0 dopo l'AND con la maschera, possiamo tirare un sospiro di sollievo e tornare alla lettura del file. Altrimenti, abbiamo rilevato un CTRL-C e dobbiamo fermare il programma.
Si noti come ciò sia elegante... lo stop può avvenire solo alla fine di una linea. Nessuna linea verrà interrotta a metà. All'inizio della nuova linea, stamperemo un "Control-C" composto da "^","C" e NewLine per segnalare l'interruzione. Poi ci fermeremo.

```
move.l  a5,d1              ; Filehandle
lea     CtrlCMess(pc),a0 ; Messaggio da stampare
move.l  a0,d2              ; Buffer
moveq   #3,d3              ; Lunghezza
jsr     _LVOWrite(A6)
```

Sembra strano, ma mettere uno 0 nel registro D5 fermerà il programma. Lo spiegheremo fra un attimo.

```
  moveq   #0,d5         ; Azzerare D5 fa concludere
NotCtrlC
  moveq   #0,d7         ; Contatore delle colonne=0
  bra.s   NextChar
```

Arriviamo alla routine EndBuf quando abbiamo esaurito il buffer e dobbiamo riempirlo nuovamente, in modo da continuare a leggere il file. Come sapere quando fermarsi? C'era un indizio nella nostra discussione di Read.

Se non siamo alla fine del file, il buffer viene riempito. Solo alla fine del file il buffer risulta parzialmente vuoto. Ora, il numero dei caratteri letti era stato posto in D5. Se corrisponde alla lunghezza del buffer possiamo tornare indietro e ricominciare. Altrimenti, il file è terminato. Stampiamo un carattere di NewLine per concludere con eleganza.

Esercizio mentale: che cosa accade se l'ultimo blocco riempie esattamente il buffer?

```
  ; Il buffer è esaurito. Se era pieno, riempiamolo
  ; nuovamente.
EndBuf
  cmp.w   #BufSize,d5  ; Era un buffer pieno?
  beq     ReadBlock    ; Sì, riempiamolo di nuovo
  move.b  #$0a,(a3)    ; No, inviamo NewLine
  move.l  a5,d1        ; Filehandle
  move.l  a3,d2        ; Buffer
  moveq   #1,d3        ; Lunghezza
  jsr     _LVOWrite(A6)
  rts

dosname    dc.b 'dos.library',0
CtrlCMess  dc.b '^C',$0a
```

I nostri dati fissi sono esattamente qui sopra: il nome della libreria DOS e il messaggio per CTRL-C. Veniamo ai seguenti: i nostri due buffer. Quello di output è lungo solo un carattere, quello di input dieci. "DC.B 0" significa "metti qui uno 0"; "DS.B 10" significa "prevedi lo spazio per dieci caratteri, non ha importanza che cosa contengano".

```
OutBuf    dc.b  0
BufSize   EQU   10
BufAdr    ds.b  BufSize
```

La direttiva assembler EQU (EQUate, equivale) non genera codice od occupazione di memoria.
Semplicemente significa "tutte le volte che incontri BufSize, attribuiscigli il valore 10".
E' tutto. Assemblatelo e linkatelo, e avrete un piccolo e maneggevole strumento per indagare sul contenuto dei file.

Pensieri impuri

Il codice che abbiamo visto non è puro. Se il programma tentasse di fare due (o più) operazioni alla volta, i due buffer probabilmente mescolerebbero i dati.
Ciò non accadrebbe se noi aprissimo due finestre CLI distinte e in ciascuna comandassimo DUMP <nome_del_file>. Per ciascuna finestra sarebbe caricata una copia separata del programma DUMP e non ci sarebbe alcun conflitto. Il conflitto emergerebbe, invece, se rendessimo DUMP residente.
Assumendo che voi abbiate il Workbench 1.3, tentate il comando RESIDENT DUMP PURE (PURE permette di superare il fatto che non è stato impostato il bit "pure" nel file del programma. Ora, apriamo due finestre Shell... solo le fine-

Istruzioni incontrate fino a questo momento

ADD <ea>,Dn	Somma binaria	[.B,.W,.L]
ADD Dn,<ea>	Somma binaria	[.B,.W,.L]
ADDA <ea>,An	Somma a un registro indirizzi	[.W,.L]
ADDI #<dato>,<ea>	Somma con un valore immediato	[.B,.W,.L]
ADDQ #<dato>,<ea>	Somma veloce	[.B,.W,.L]
AND <ea>,Dn	AND logico	[.B,.W,.L]
AND Dn,<ea>	AND logico	[.B,.W,.L]
ANDI #<dato>,<ea>		[.B,.W,.L]
BEQ	Salta se uguale	[opz..S]
BNE	Salta se diverso	[opz..S]
BGE	Salta se maggiore o uguale (con segno)	[opz..S]
BLT	Salta se minore (con segno)	[opz..S]
BGT	Salta se maggiore (con segno)	[opz..S]
BLE	Salta se minore o uguale (con segno)	[opz..S]
BRA	Salta sempre	[opz..S]
BSR	Chiama la subroutine	[opz..S]
CLR <ea>	Azzera	[.B,.W,.L]
CMP <ea>,Dn	Confronta	[.B,.W,.L]
CMPA <ea>,An	Confronta con un registro indirizzi	[.W,.L]
CMPI #<dato>,<ea>	Confronta con un valore immediato	[.B,.W,.L]
JSR <ea>	Chiama la subroutine	
LEA <ea>,An	Carica l'indirizzo	[.L]
LINK An,#<scostamento>	Riserva un'area di memoria sullo stack	
MOVE <ea>,<ea>	Muove dati	[.B,.W,.L]
MOVEA <ea>,An	Muove dati in un registro indirizzi	[.W,.L]
MOVEQ #<dato>,Dn	Muove velocemente	[.L]
RTS	Ritorno da subroutine	
SUB <ea>,Dn	Sottrazione binaria	[.B,.W,.L]
SUB Dn,<ea>	Sottrazione binaria	[.B,.W,.L]
SUBA <ea>,An	Sottrazione da registro indirizzi	[.W,.L]
SUBI #<dato>,<ea>	Sottrazione immediata	[.B,.W,.L]
SUBQ #<dato>,<ea>	Sottrazione veloce	[.B,.W,.L]
TST <ea>	Test, confronta con 0	[.B,.W,.L]
UNLK An	Libera la memoria dello stack	

<ea> (effective address, indirizzo effettivo) indica qualsiasi modo di indirizzamento

Tabella 1

stre Shell usano i comandi residenti. In ciascuna finestra digitiamo DUMP <nome_del_file>.

I nomi dei file possono essere identici o differenti; suggerisco di usare file di testo in RAM:, come i vostri file sorgenti .ASM. Sono facili da leggere, e potrete focalizzare qualsiasi problema rapidamente.
Una sola copia di DUMP in memoria tenta di esaminare entrambi i file e, come potrete vedere, fallirà miseramente. Non esiste pericolo per il sistema in questo caso, ma potrete vedere che l'output è un caos.
Sono solo i buffer a causare questo effetto. C'è un rimedio semplice? Ricavare i nostri buffer dallo stack.

LINK e UNLK

Se il programma contiene LINK A4, #-$10, 16 byte (10 esadecimale) saranno ricavati dallo stack e messi a vostra disposizione. Il registro A4 punterà alla sommità di questa area di memoria; possiamo indirizzarla utilizzando un offset rispetto ad A4.

Dobbiamo assolutamente restituire la memoria quando abbiamo terminato, con UNLK A4. A causa del modo in cui funziona la memoria dello stack, dobbiamo restituire la memoria nella stessa subroutine che ne ha fatto richiesta. Una spiegazione dettagliata di come lavorano i registri e lo stack, con i comandi LINK e UNLK, porterebbe via troppo spazio. Basti per ora sapere che è un modo maneggevole e sicuro per ottenere piccole quantità di memoria. Si veda la figura 1 per una spiegazione di che cosa succede quando li si usa.

Un programma puro

Ciò che segue è simile alla precedente versione di DUMP. Solo il codice nuovo verrà commentato.

```
      ; Dump puro - Jim Butterfield. Marzo 16/89.
      ; Funzioni della libreria Exec

      xref    _LVOOpenLibrary  ;-$228
      xref    _LVOCloseLibrary ;-$19E
      xref    _LVOSetSignal    ;-$132
      ; Funzioni della libreria DOS
      xref    _LVOOpen         ;-$1E
      xref    _LVOClose        ;-$24
      xref    _LVOOutput       ;-$3C
      xref    _LVOWrite        ;-$30
      xref    _LVORead         ;-$2A

   BufSize EQU 10

      ;-- Inizializzazione:

Startup
      move.l  a0,a4 ; Puntatore alla linea di comando
      move.l  d0,d4        ; Lunghezza
```

```
     lea     dosname(pc),a1 ; 'dos.library'
     clr.l   d0           ; Qualsiasi versione
     move.l  $4,a6        ; ExecBase
     jsr     _LVOOpenLibrary(a6) ; Apriamo il DOS
     move.l  d0,a6   ; Salviamo il puntatore al DOS
     tst.l   d0    ; Controlliamone il valore
     beq.s   StartupQuit ; Usciamo in caso di errore
     bsr.s   DOSinit      ; Eseguiamo il lavoro
     move.l  a6,a1   ; Puntatore al DOS in a1
     move.l  $4,a6        ; ExecBase
     jsr     _LVOCloseLibrary(a6) ; Chiudiamo il DOS
  StartupQuit

     rts              ; Fine del programma
     ;
     ;-- Ricaviamo l'handle per l'output del CLI:

DOSinit
     jsr     _LVOOutput(a6) ; Ricaviamo l'handle di
     output
     move.l  d0,a5        ; e salviamolo

     ;-- Trasformiamo il nome del file nella linea di
     comando
     ;-- in una stringa C:
  checklen
     move.b  #0,-$1(a4,d4.W) ; Poniamo uno 0 binario
     in fine
  skipspc
     cmp.b   #$20,(a4)+
     beq.s   skipspc
     subq.l  #1,a4
     move.l  a4,d1 ; Puntatore al nome del file
     move.l  #1005,d2     ; MODE_OLDFILE (per la
     lettura)
     jsr     _LVOOpen(A6)
     move.l  d0,d6    ; Filehandle per l'input
     beq.s   DOSquit      ; Errore, si termina
```

Proprio prima di svolgere il lavoro fondamentale, chiediamo allo stack di fornirci 10 byte (esadecimale).

```
     link    a4,#-$10
     bsr.s   DumpFile     ; La sezione fondamentale
```

Restituiamo i byte, presi a prestito, allo stack.

```
     unlk    a4
     move.l  d6,d1        ; Usiamo il filehandle...
     jsr     _LVOClose(A6) ; per chiudere il file
  DOSquit
     rts                  ; Ritorniamo.
     ;
     ; Il DOS è aperto e abbiamo gli handle di input
     e output.
```

Modi di indirizzamento incontrati fino a questo momento

Dn	**Registro dati diretto** Il contenuto del registro dati.
An	**Registro indirizzi diretto** Il contenuto del registro indirizzi
(An)	**Indiretto da registro indirizzi** Il contenuto della memoria indirizzata dal registro indirizzi
(An)+	**Indiretto da registro indirizzi con incremento successivo** Come sopra, ma il registro indirizzi è incrementato dopo l'uso
-(An)	**Indiretto da registro indirizzi con decremento precedente** Come (An), ma il registro viene decrementato prima dell'uso.
x(An)	**Indiretto da registro indirizzi con scostamento** Il contenuto della memoria indirizzata da: il registro indirizzi più uno scostamento compreso fra -32768 e +32767.
x(An,Dm)	**Indiretto da registro indirizzi con scostamento e indice** Il contenuto della memoria indirizzata da: il registro indirizzi, più l'indirizzo dati (word o longword), più uno scostamento compreso fra -32768 e +32767.
x	**Assoluto** L'indirizzo indicato nel programma.
x(PC)	**Indiretto da Program Counter (PC) con scostamento** La "locazione corrente" del programma, più uno scostamento compreso tra -32678 e 32676.
x(PC,Rn)	**Indiretto da Program Counter (PC) con scostamento e indice** La "locazione corrente" del programma, più il registro dati o indirizzi (word o longword), più uno scostamento compreso tra -128 e 127.
#x	**Immediato** Un valore reale da utilizzare.

Tabella 2

```
    ; Partiamo!
DumpFile
    moveq   #0,d7         ; Azzeriamo il contatore
ReadBlock
    move.l  d6,d1         ; Filehandle di input
```

Si noti come gli indirizzi dei buffer vengano impostati in maniera analoga al programma precedente, tranne per il fatto che si fa uso di un offset rispetto ad A4.

```
    lea  -$10(a4),a2  ; Buffer di input (stack)
    lea  -$4(a4),a3   ; Buffer di output (stack)
    move.l  a2,d2   ; Copiamolo in D2 per Read
    moveq   #BufSize,d3 ; Lunghezza del buffer di
    input
    jsr     _LVORead(A6) ; Leggiamo il file
    move.l  d0,d5      ; Lunghezza dei dati letti

    ; Qui operiamo l'analisi dei caratteri e
    l'output
    moveq   #0,d4         ; Inizio del buffer

    ; Si noti che A2 contiene ancora l'indirizzo del
    buffer
    ; di input
    ; Loop dei caratteri - siamo alla fine del
    buffer?
NextChar
    cmp.l   d5,d4
    bcc.s   EndBuf
    ; Abbiamo i dati! Elaboriamoli.
    move.b  $0(a2,d4),d0
    addq.l  #1,d4
    cmp.b   #$0a,d0
    beq.s   NuLiner
    cmp.b   #$1F,d0
    bgt.s   NotDot
    move.b  #$2e,d0
NotDot
    move.b  d0,(a3)
    move.l  a5,d1         ; Filehandle
    move.l  a3,d2         ; Buffer di output
    moveq   #1,d3         ; Lunghezza
    jsr     _LVOWrite(A6)
    ; Calcoliamo e inviamo NewLine se necessario
    addq.b  #1,d7       ; Aggiorniamo il contatore
    cmp.b   #64,d7        ; Siamo al limite?
    bls.s   NextChar
NuLiner
    move.b  #$0a,(a3)
    move.l  a5,d1         ; Filehandle
    move.l  a3,d2         ; Buffer
    moveq   #1,d3         ; Lunghezza
    jsr     _LVOWrite(A6)
    move.l  a6,d7         ; Salviamo DOSBase
    moveq   #0,d0
    move.l    #$1000,d1
    movea.l   $4,a6
    jsr _LVOSetSignal(A6) ; Testiamo CTRL-C
    move.l    d7,a6      ; Recuperiamo DOSBase
    andi.l    #$1000,d0
    tst.l d0
    beq.s     NotCtrlC
    move.l    a5,d1     ; Filehandle
    lea CtrlCMess(pc),a0 ; Messaggio da
                               stampare
    move.l    a0,d2        ; Buffer
        moveq     #3,d3        ; Lunghezza
        jsr       _LVOWrite(A6)
        moveq     #0,d5     ; Azzerare D5 fa concludere
NotCtrlC
        moveq     #0,d7  ; Contatore delle colonne=0
        bra.s     NextChar
        ; Il buffer è esaurito. Se era pieno, riempiamolo
        ; nuovamente
EndBuf
        cmp.w     #BufSize,d5  ; Era un buffer pieno?
        beq       ReadBlock  ; Sì, riempiamolo di nuovo.
        move.b    #$0a,(a3)    ; No, inviamo NewLine
        move.l    a5,d1        ; Filehandle
        move.l    a3,d2        ; Buffer
        moveq     #1,d3        ; Lunghezza
        jsr       _LVOWrite(A6)
        rts                    ; Lavoro concluso

dosname   dc.b  'dos.library',0
CtrlCMess dc.b  '^C',$0a

end
```

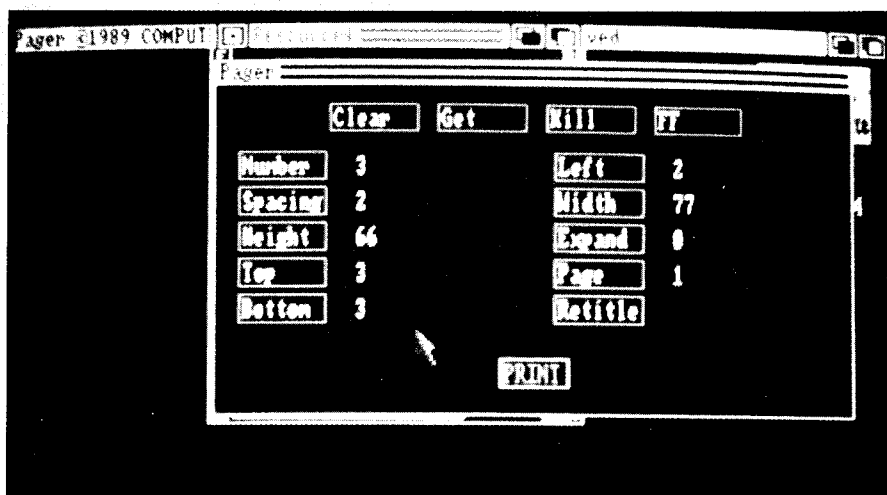
Provate a usare questo, come programma residente. Scoprirete che è puro come la neve accumulata dal vento.

tenendo premuto lo shift su tutte le altre e, infine, sull'icona di V (tenendo sempre premuto lo SHIFT).

3) Modificando il tool di default (Default tool) della figura da visualizzare (cliccate sull'icona della figura una sola volta, scegliete l'opzione "info" del menu "WorkBench", e scrivete all'interno del requester "Default Tool" il nome dell'utility:

V ; selezionate poi in gadget Save ; da quel momento il poi cliccando due volte sull'icona della figura, verrà automaticamente richiamata l'utility V che la visualizzerà).

4) Se nella vostra directory LIBS è presente "arp.library", potete semplicemente cliccare sull'icona di V, e un comodo menu vi permetterà di scegliere la figura da visualizzare muovendovi eventualmente in directory e sottodirectory di un qualsiasi device di memorizzazione.
Cliccando sul gadget DRIVES verranno visualizzati tutti i device connessi al sistema, e per muoversi in uno dei device (o in genere in una directory) basta cliccarci sopra con il tasto sinistro, così come per selezionare il file da visualizzare.

Durante la visualizzazione dell'immagine i tasti cursore permettono di muoverla a piacimento all'interno dello schermo.
E' da notare che le immagini in HiRes interlacciato potrebbero sfarfallare più del normale in determinate posizioni: questo è dovuto ad alcune limitazioni hardware di Amiga, e non al programma; il difetto è eliminabile spostando l'immagine (anche di pochi pixel).
Per spostare immagini che non possono essere completamente visualizzate all'interno dello schermo, tenete premuto il tasto SHIFT mentre utilizzate i tasti cursore; questa operazione permette di muovere l'immagine al di là dei limiti dello schermo.
Quando avete terminato di osservare la figura premete il tasto sinistro del mouse e il controllo tornerà al WorkBench.

Page

Page è un utility di stampa che possiede pressoché tutte le caratteristiche che si potrebbero desiderare da un programma del genere.
L'utility, ricevendo come ingresso un qualsiasi file ASCII creato con un word processor permette di stamparlo con varie aggiunte che lo rendono meglio leggibile, oltre a rendere possibile una pratica gestione dei parametri di stampa quali la lunghezza della pagina, delle righe, la spaziatura, i margini di inizio e fine pagina, eccetera. A Page, l'utility vera e propria che elabora il testo e lo invia alla stampante o allo standard output, si aggiunge Pager, che parmette di creare, attraverso un'interfaccia di tipo intuition, il comando da inviare a Page per ottenere un certo setting dei parametri.
Utilizzando da CLI Page è, invece, necessario indicare con vari caratteri di controllo seguiti da valori numerici i vari parametri di stampa; eccone i significati:

C (Clear) Evita che venga mandata la linea di intestazione in fase di stampa (ossia la linea contenente il titolo, la data, l'ora e il nome del file).
Se non viene indicata l'opzione C tale linea viene stampata all'inizio di ogni pagina.

G (Get permission) Permette di fare in modo che Page attenda la pressione di un tasto prima di iniziale la stampa di ogni pagina: premendo RETURN si dà il via alla stampa, premendo Esc e RETURN si annulla la stampa.
Se si redireziona con il carattere > l'output del testo questa opzione non funziona.

K (Kill) Tronca le linee troppo lunghe invece di mandarle a capo.
F (Form Feed) Dice a Page di mandare un form feed dopo la stampa di un documento (utile se si utilizzano Wildcard per stampare uno dopo l'altro più documenti).

N (number) Numera le linee stampate (utile per stampare listati di programmi per individuare le linee che il compilatore indica come errate). Un parametro opzionale che può seguire N indica il numero di caratteri riservati alle cifre (3 di default)

S (spacing) Serve a scegliere la spaziatura fra una linea e la successiva.
Se non viene indicata l'opzione, la spaziatura è singola, altrimenti il numero che segue la S indica il fattore di spaziatura (default 2).

H (Height) Modifica l'altezza totale in linee della pagina (compresi i margini superiore e inferiore e la linea di intestazione, se c'è). Default 66 linee

T (top) Indica l'altezza in linee del margine superiore (senza opzione indicata il default è 3, con l'opzione non seguita da valore è 0).

B (Bottom) Idem per il margine inferiore

L (left) Idem per il margine sinistro (senza opzione il default è 2 e non 3).

W (Width) Indica la lunghezza delle linee (senza opzione il valore di default è 77, indicando l'opzione è obbligatorio indicare anche il valore associato.)

X (eXpand) Modifica le posizioni di tabulazione.
Senza indicare l'opzione il valore di default è un Tab ogni quattro caratteri.
Indicando l'opzione senza aggiungere un valore i Tab vengono ignorati, così come vengono ignorati facendo seguire uno 0 alla X. Dando un altro valore "n" come parametro i Tab saranno posizionati ogni n caratteri.

P (Page) Permette di iniziare la stampa a partire da una determinata pagina (è obbligatorio fornire il valore numerico relativo alla pagina).

R (Retitle) Rimpiazza il titolo di default con un titolo personalizzato, formato da un testo e dal numero di pagina.

L'opzione può essere data in cue diverse forme:

-Rtesto costruisce un intestazione formata dal testo seguito da uno spazio e dal numero di pagina.

-R testo costruisce la stessa intestazione, ma senza spazio permettendo di separare testo e numero di pagina con un carattere posto alla fine del testo (un trattino ad esempio).

Come vanno indicate le opzioni

Page è molto versatile poiché permette di indicare le opzioni e il nome del file in qualsiasi ordine (le seguenti tre linee hanno lo stesso effetto, ad esempio)

```
page -GW120CK page.c
page page.c -k -gw 120 -c
page -cW120 page.c -kg
```

Si nota come le opzioni o i gruppi di opzioni sono preceduti dal trattino.
E' possibile redirezionare l'output ad una periferica diversa dalla stampante con il carattere " > " (nel qual caso il nome del file di ingresso va preceduto dal simbolo " < ").

Il programma Pager permette di evitare la fatica di indicare i vari parametri scegliendo le varie opzioni tramite icone permettendo, inoltre, di salvare particolari configurazioni su disco.
Con l'icona PRINT si manda in stampa il documento.

Hotkey!

Capita molto spesso di dover far partire programmi che si utilizzano molto spesso come, ad esempio, il CLI.
HotKey permette di far partire fino ad 11 programmi a scelta anche se il CLI non è attivo, premendo il tasto Amiga Sinistro assieme ad ESC o ad uno dei dieci tasti funzione.
Inizialmente i vari tasti hanno già assegnati dei programmi o dei comandi CLI, ma è possibile modificare il settaggio di HotKey mediante SetKey, un programma che è facilmente richiamabile premendo Amiga-Sinistro assieme a F-10.
Questo programma permette di scrivere in undici requester le 11 chiamate a programmi o comandi CLI da associare ad ogni combinazione e salvare i nuovi setting per poi utilizzarli con HotKey.
E' importante indicare esattamente il path da seguire per raggiungere ogni programma o comando CLI in quanto altrimenti HotKey non sarà in grado di rintracciarlo.
Quando HotKey è attivo, premendo il tasto Amiga-Sinistro assieme al tasto DEL è possibile disattivarlo.

ATTENZIONE :
I programmi Mackie (rubrica Public Domain) e Access (Speciale Comunicazione) sono registrati nel cassetto UTILITIES

AmigaTEX, ovvero l'essenza del WYSINWYG

Siamo lieti di presentare ai nostri lettori la recensione di un super programma per impaginare !

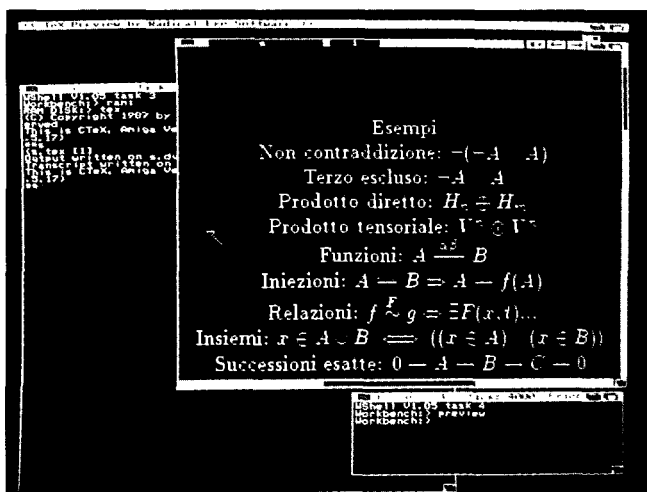
Sebastiano Vigna

Probabilmente poche persone, tra quelle che stanno leggendo queste righe, conosceranno TeX, e probabilmente un numero ancora minore sarà a conoscenza della persona e dell'opera di D.E. Knuth. Si rende, quindi, necessaria una piccola digressione.

D.E. Knuth è probabilmente il più importante studioso di informatica vivente. Chiunque abbia avuto a che fare con argomenti affini conoscerà la sua celebre opera The Art of Programming, che raccoglie e discute centinaia di algoritmi di tutti i generi (numerici e non). Contrariamente però ad altri teorici, Knuth è anche un grande pratico, e una delle sue realizzazioni maggiori rimane senza dubbio il sistema TeX, descritto completamente in più libri pubblicati sotto il nome Computers & Typesetting. In particolare, i due testi Metafont: The program e TeX. The program definiscono, in un linguaggio derivato dal Pascal, una coppia di programmi destinati a lavorare sinergicamente alla produzione di testi stampati. Poiché le idee dietro a questi due programmi sono da una parte rivoluzionarie, ma dall'altra completamente controcorrente rispetto all'andamento attuale dell'industria informatica, mi è parso giusto dedicare loro un po' di spazio.

Metafont è un programma che permette di generare font di qualunque dimensione e tipo sulla base di una loro descrizione effettuata in uno specifico linguaggio, ed esegue questa operazione in termini di primitive matematiche, esattamente come il PostScript. La differenza fondamentale è che, mentre quest'ultimo deve lavorare in tempo reale, ovvero in fase di stampa, Metafont genera una volta per tutte i caratteri desiderati sotto forma di bitmap quando sia specificata la densità della stampante. Le bitmap così ottenute sono poi a loro volta utilizzate dal programma che potremmo chiamare principale, ovvero TeX, che non si legge "tecs", bensì "tec", in quanto si tratta delle lettere

E' possibile far girare il previewer nello schermo del Workbench.

Un esempio del massimo ingrandimento disponibile; il font visualizzato è da 10 punti.

greche "tau\epsilon\chi" scritte in carattere maiuscolo. Secondo Knuth, questa specie di gioco verbale serve a ricordare che il programma si prefigge di stampare testi di qualunque tipo, ma in particolare tecnico/matematici, tenendo sempre presente il risultato estetico: nel caso non lo sapeste, la radice greca alla base di parole come "tecnologia" o "tecnica" significa anche "arte". "Se volete solo produrre un documento passabilmente buono (qualcosa di accettabile e di fondamentalmente leggibile ma non veramente bello) un sistema più semplice di solito basterà. Con TeX l'obiettivo è quello di produrre la migliore qualità" (D.E. Knuth, The TeXbook, Addison-Wesley, p. 1, traduzione mia).

Oggi esistono implementazioni di TeX per quasi tutti i computer, micro e mini. In particolare, esiste una versione per Amiga, che, oltre ad offrire la normale gamma di possibilità, grazie alla grafica e al multitask aggiunge una dimensione in più all'intero sistema. AmigaTeX trae la sua origine dal port verso UNIX di TeX, detto CTeX, ottenuto rielaborando il programma in C, operazione resa necessaria dalla mancanza di buoni compilatori Pascal in quell'ambiente.

TeX lavora in modo simile ad un compilatore. Accetta in ingresso un file di testo e produce un file di estensione ".dvi", assimilabile ad un file oggetto. La sigla sta a significare DeVice Independent, in quanto il formato di uscita di TeX è indipendente dalla periferica di output. Sarà il driver di stampa a mettere insieme le bitmap prodotte da Metafont e i file .dvi, in modo da ottenere la stampa finale.

Uno dei grandi meriti del programma è quello di essere divenuto uno standard a livello mondiale. Ogni implementazione prima di potersi chiamare TeX deve essere certificata passandole un file elaborato da Knuth, chiamato TRIP. Se l'output in formato .dvi generato dal programma è identico a quello dell'originale, il programma è considerato conforme allo standard a tutti gli effetti. Il file .dvi generato potrà essere stampato su una laser, su una stampante a matrice o su una Linotronic da 1200 punti per pollice senza alcuna modifica. Ovviamente l'aspetto dipenderà dalla risoluzione, ma le dimensioni e la disposizione del materiale saranno gli stessi, in qualunque modo otteniate il vostro output.

Quel che rende TeX così controcorrente è la sua filosofia che, ironicamente, ho chiamato WYSINWYG (What-You-See-Is-Not-What-You-Get): il file che accetta in ingresso è codificato con speciali convenzioni, a volte molto complesse e, fino al momento della stampa, non c'è modo di farsi un'idea del risultato. O meglio, non c'era.

La Radical Eye Software, produttrice del pacchetto, ha ben pensato di utilizzare la struttura multitask di Amiga per dare un tocco di grafica al programma. Il previewer di AmigaTeX può mostrare in alta risoluzione e a più ingrandimenti un file .dvi,

Un esempio di stampa musicale con MTeX.

Beispiele für den Notensatz mit TeX — Schofer/Steinbach 1987

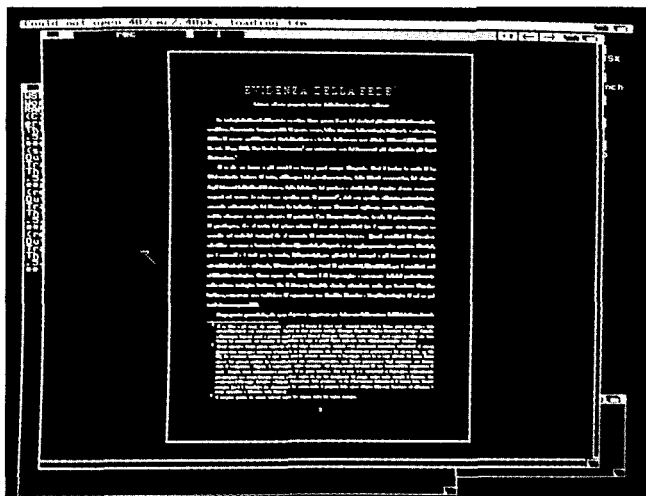
6 Suiten für Cello Solo
Suite II Menuet I

Für Viola übertragen
von Franz Schmidtner

Johann Sebastian Bach

Aus: Johann Sebastian Bach, Sechs Suiten, für Viola bearbeitet von Franz Schmidtner, Musikverlag Hans Sikorski, Hamburg.

Für dieses Musikstück lautet die Eingabe:

```
!! \title{6 Suiten für Cello Solo \atop Suite II Menuet I}
\composer{Johann Sebastian Bach} \poet{Für Viola übertragen \atop von Franz Schmidtner}
\alto\F\meter{3/4}\two
{   \chord{2'd f a} 4h|\chord{'8c 'e}/ / / 4a|
\chord{4,h 'd} g\beam{8f e}|,8a / / / / / |\chord{'2d f a} 4h|8e / / / "4c |
\chord{4'f a}\chord{,h 'f "d} \chord{,g 'g "e}|
\chord{2.,a ,'e ,x"c} :||:
\chord{2'a "e}\slur{\beam{8d 6c f}}|8'g/ / / / / |2f / 4e|8d / / / / /|\slur{4g f}e|
8f/ / / \chord{4,h 'e "d}|\chord{4,a 'f "c} \beam{8'h a g a}|
\beam{8f e f a g h}|2xf 4p |b8e / / / 4xf \tr|\chord{4,h 'g "d} \chord{4'c g "c} 'd/ |
\beam{\chord{,8g 'd "c} 8'h a h g h}|4g p nf|8xc / / / 4d|
\beam{8,g 'g} 4.,a 8p| / :||  }
{  / / |\beam{\slur{8'h a} h g}/ |/ / / / |\beam{\slur{'8f e d}xc ,nh a}|/ / |
\beam{\slur{8h a}\tenuto h f \noaccent / }|/ / / |/ :||:
/ / / / |\beam{\slur{8"e d xc e 'a}g}|\slur{'4a "d}xc|\beam{\slur{8"g f e f d }nc}|
'2h/ "4c|\beam{8'a g a f}/ |/ / / / /|/ / / / / /|
2'a \tr 4h|\beam{\slur{8g f} g e}/|/ / \beam{\slur{8h a}}|/ / / / / / |
\slur{"4d 'a} "d|\beam{\slur{8'g f} g e} 4f|/ / 4.e \tr 8d|\chord{2.,d .a '.d} :||  } !!
```

Utilizzando l'ingrandimento "full" è possibile farsi un'idea dell'aspetto complessivo di una pagina.

Anche le equazioni sono perfettamente visualizzate.

e può addirittura farlo mentre TeX lo sta generando, grazie al sofisticato sistema di passaggio dei messaggi di Exec. Il risultato è un sistema che, pur mantenendo intatta la potenza del programma originario, ne rende l'uso infinitamente più facile ed intuitivo.
Non cercherò certo di spiegare nello spazio di un articolo il funzionamento di TeX, come del resto non cerca di farlo il manuale di AmigaTeX; chi fosse interessato può trovare ogni informazione nel libro di Knuth citato all'inizio dell'articolo, il celeberrimo TeXbook, ormai giunto alla quattordicesima edizione.
Essendo (o meglio, dovendo diventare) un matematico non posso nascondere che il mio interesse per TeX è anche motivato dal fatto che per stampare testi scientifici semplicemente non esiste nulla che possa reggere il paragone. Se però dovete scrivere solo lettere d'affari, può darsi che non valga la pena di fare la fatica di imparare ad usare tutto il sistema.

La prima cosa che è necessario sapere, per usare TeX, sono alcune semplici regole di sintassi: per esempio, le virgolette vanno indicate con due apici di senso diverso all'inizio e alla fine della parola. In effetti, l'appiattimento dei codici ASCII, che specificano un solo carattere per le virgolette, ha conseguenze negativissime dal punto di vista dell'estetica del testo. Allo stesso modo, TeX rende disponibili quattro trattini di lunghezza, altezza e spessore diversi: quello tra parole, quello tra numeri, quello tra frasi e quello matematico, e vengono specificati rispettivamente con "-", "--", "---" e "$-$". Al di là di questi semplici accorgimenti, TeX si occuperà automaticamente delle legature, vale a dire di quei gruppi di lettere che per ragioni estetiche vanno stampati in modo diverso, in particolare "fi","fl", "ffi", "ffl" e "ff". A tutt'oggi nessun programma di DTP supporta pienamente le legature, anche perché nel set di caratteri PostScript sono presenti le legature "fi"ed "fl", ma non le altre, e non c'é niente di peggio di un "ffi" metà legato e metà no. Una delle ragioni per cui la qualità dei libri stampati va degenerando è precisamente il fatto che tutti questi particolari non sono in genere trattati correttamente dai programmi usati in editoria.
L'altro grande punto di forza di TeX è il suo algoritmo per spezzare un paragrafo in linee. Tutti i programmi attualmente in circolazione lavorano in modo piuttosto stupido, troncando una linea ed andando a capo quando una parola non trova spazio sufficiente. L'algoritmo di Knuth (tra l'altro oggetto di una pubblicazione a parte), da lui giudicato la parte più interessante del programma, vede un paragrafo come un'unica entità ed assegna punteggi positivi o negativi a tutte le situazioni grafiche possibili: per esempio, se due linee consecutive hanno un trattino di sillabazione sul margine destro o se hanno un aspetto molto diverso a causa della giustificazione viene assegnato un punteggio negativo contenuto nelle variabili doublehyphensdemerits e adjacentpenalty, regolabili dall'utente.
Il risultato è l'uniformità assoluta nell'output, anche su un numero estremamente ristretto di colonne. TeX non cercherà mai di inserire microspazi tra le lettere di una parola (orrore tipografico su cui molti programmi di DTP infieriscono) bensì cercherà di trovare la disposizione del paragrafo a cui corrisponde, a conti fatti, la penalità più bassa. Se le cose vanno così male da rimanere sempre sopra il livello specificato in tolerance, allora darà un messaggio di errore.
Dopo questo excursus all'interno del programma, andiamo a parlare di AmigaTeX. La novità essenziale rispetto al programma di Knuth è data dal previewer, che permette di visualizzare in alta risoluzione un file .dvi. Sono disponibili molti livelli di

ingrandimento, e più di 1500 font su disco. Nel caso un certo font non sia disponibile, vengono visualizzati al suo posto dei blocchi neri le cui dimensioni danno un'idea approssimativa dell'output finale. I menu del programma sono ricchi di opzioni, tra cui quella di aprire la finestra di preview nello schermo del Workbench oppure in uno proprio.
L'istruzione forse più interessante è quella di Track, che nel caso TeX sia caricato in memoria dice al previewer di "agganciarsi" al suo output e di visualizzarlo, in tempo reale, proprio mentre viene generato.
Ovviamente, è possibile in un dato momento esaminare soio le pagine che TeX ha finito di elaborare, eventualmente accoppiando a due a due quelle che si troveranno faccia a faccia.
Il funzionamento è estremamente intuitivo e la risposta del programma assolutamente immediata. Una lode particolare va fatta all'implementazione asincrona dell'I/O rispetto all'aggiornamento video: in poche parole, si può cambiare pagina, testo, ingrandimento mentre il previewer aggiorna lo schermo.
In tal caso, interromperà quello che stava facendo e riprenderà seguendo le nuove indicazioni; fino ad oggi, questo è l'unico programma per Amiga da me visionato che offra qualcosa del genere.
Una misura dell'affidabilità e della potenza del prodotto può essere data dal fatto che ogni giorno cinquanta Amiga allo SLAC (Stanford Linear Accelerator Center) permettono ai ricercatori di controllare i loro testi in formato TeX prima di stamparli (la versione per mainframe non possiede un previewer).
Ad AmigaTeX sono allegate diverse utility, tra cui una, molto interessante, che permette di trasformare i font dal formato .pk (packed) nel formato .font standard di Amiga.
In poche parole, i bellissimi font disegnati da Knuth possono essere utilizzati con qualunque programma che acceda ai font di sistema, da Excellence! a DeluxePaint, anche se ovviamente non è possibile ottenere finezze tipografiche come le legature. Un'altra utility provvede, invece, a trasformare un file sorgente per AmigaTeX in uno standard TeX.
La conversione si rende necessaria perché la Radical Eye Software ha ben pensato di implementare una traduzione automatica dei caratteri accentati di Amiga nel formato con cui normalmente andrebbero scritti (ad esempio, à si dovrebbe scrivere \`a). Questo fatto può però causare incompatibilità con TeX funzionanti su altre macchine, ad esempio su mainframe: di qui la necessità di una utility di conversione.

Con legature: ﬁ ﬂ ﬃ ﬄ ﬀ
Senza legature: fi fl ffi ffl ff
Con kerning: AVAW Po ATTo
Senza kerning: AVAW Po ATTo

Aperte virgolette: “ **Chiuse virgolette:** ” **Trattini:** - – —

Alcuni font:
ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
αβγδεεηθϑζικλμνξπϖρϱσςτυφϕχψω

Esempi
Non contraddizione: $\neg(\neg A \wedge A)$
Terzo escluso: $\neg A \vee A$
Prodotto diretto: $H_n \oplus H_m$
Prodotto tensoriale: $V^n \otimes V^n$
Funzioni: $A \xrightarrow{\alpha\beta} B$
Iniezioni: $A \hookrightarrow B \Rightarrow A \leftrightarrow f(A)$
Relazioni: $f \overset{F}{\sim} g \iff \exists F(x,t)\ldots$
Insiemi: $x \in A \cup B \iff (x \in A) \vee (x \in B)$
Successioni esatte: $0 \to A \to B \to C \to 0$
Somme:

$$e(i) = \sum_{n=0}^{i} \frac{x^n}{n!}$$

Integrali:

$$\int_a^b e(x)dx = [e(x)]_a^b = e(b) - e(a)$$

Legature, trattini ed equazioni.

AmigaTeX merita anche una menzione particolare per essere stato uno dei primi, e probabilmente il primo programma per Amiga a supportare ARexx. Sia il previewer che TeX, dispongono di porte separate attraverso cui è possibile pilotarli. Ad esempio, ho installato nel CygnusEd un file "TeXit.ced" che automaticamente salva il testo che

sto scrivendo e dice ad AmigaTeX di impaginarlo. Il previewer, se presente, balza in questo caso automaticamente davanti agli altri schermi... provate a farlo su qualunque altro sistema! Si parla molto oggi di "software modulare", ma solo su Amiga strutture di questo tipo sono già state pienamente realizzate.
Al pacchetto standard sono anche allegati LaTeX, BibTeX e AMSTeX, tre insiemi di macro orientati rispettivamente alla produzione di documenti, alle bibliografie e alla stampa di testi matematici. Come se ciò non bastasse, la Radical Eye può fornire, per un costo nominale, MTeX, ancora un insieme di macro che permette la stampa musicale più raffinata che abbia visto su Amiga.
Credo che AmigaTeX sia destinato ad un grandissimo successo. La sua situazione, in questo momento, mi fa pensare ad ARexx un paio d'anni fa. Oggi tutti pubblicano, parlano, scrivono di questo fantastico linguaggio, e la Commodore lo includerà nella 1.4, ma due anni fa era solo un ristretto gruppo di accoliti a conoscere ed apprezzare il lavoro di Bill Hawes, il che rappresenta esattamente lo stato attuale di AmigaTeX. Il fatto che la Radical Eye abbia cominciato a "spingerlo" attraverso annunci pubblicitari e simili fa comunque sperare in una maggiore diffusione di questo pacchetto, certamente uno dei più belli esistenti per Amiga. Se non altro, è certamente il sistema TeX più sofisticato disponibile sul mercato, e non sto parlando solo di micro, ma anche di mini e di mainframe.
Per procurarselo, bisogna rivolgersi direttamente alla Radical Eye, che sarà felicissima di mandare un disco di demo gratuito a chiunque glielo chieda.
Tanto per orientarsi, AmigaTex (completo di Metafont) costa $200 e il driver per una certa stampante (da specificare) $100. Quindi, con $300 diventate possessori di un sistema TeX completo (20 dischi!). Anzi, molto di più, dato che AmigaTeX supporta lo standard IFF e molti add-on di cui solo Amiga dispone. Esiste il TUG (TeX User Group) e ci sono decine di libri sull'argomento, primi fra tutti quelli di Knuth, per cui non correte certo il rischio di rimanere soli. Anche LaTeX e ASMTeX dispongono di una vasta bibliografia. Personalmente, ho abbandonato per sempre WordPerfect, Excellence! e tutti gli altri. La LasertJet IIP da diverso tempo non riceve output formattato che dal driver di AmigaTeX, e i risultati si vedono. Certamente la ripida curva di apprendimento del programma lo rende inadatto a piccoli lavori di videoscrittura, ma se quello che vi interessa è una stampa veramente fuori dal comune, e se non vi scoraggia l'idea di essere insieme scrittori e tipografi, rimarrete incantati dalle infinite possibilità di AmigaTeX. ▲

Prodotto da:
Radical Eye Software
Box 2081
Stanford, CA 94309
U.S.A.

Nella parte superiore si può vedere il file sorgente TeX che ha prodotto la stampa sottostante

```
\font\twelvefont=cmr12
\twelvefont
\font\big=cmr17 scaled\magstep2
\font\it=cmti12
\centerline{\big Amiga{\TeX}, ovvero l'essenza del WYSINWYG}
\vskip .5cm
\centerline{di Sebastiano Vigna}
\vskip .5cm
\baselineskip=15pt
\parskip=10pt plus 1pt minus 1pt
\parindent=40pt
Probabilmente poche persone tra quelle che stanno leggendo queste righe
conosceranno {\TeX}, e probabilmente un numero ancora minore sarà a
conoscenza della persona e dell'opera di D.  E.  Knuth.  Si rende quindi
necessaria una piccola digressione.

D.  E.  Knuth è probabilmente il più importante studioso di informatica
vivente.  Chiunque abbia avuto a che fare con argomenti affini conoscerà la
sua celebre opera {\it The Art of Computer Programming}, che raccoglie e
discute centinaia di algoritmi di tutti i generi (numerici e non),
Contrariamente però ad altri teorici, Knuth è anche un grande pratico, e
una delle sue realizzazioni maggiori rimane senza dubbio il sistema {\TeX},
descritto completamente in più libri pubblicati sotto il nome {\it
Computers~\& Typesetting}; in particolare, i due testi {\it Metafont:~The
program} e {\it{\TeX}:~The program} definiscono, in un linguaggio derivato
dal Pascal, una coppia di programmi destinati a lavorare sinergicamente
alla produzione di testi stampati.  Poiché le idee dietro a questi due
programmi sono da una parte rivoluzionarie, ma dall'altra completamente
controcorrente rispetto all'andamento attuale dell'industria informatica,
mi è parso giusto dedicare loro un po' di spazio.
```

AmigaTEX, ovvero l'essenza del WYSINWYG

di Sebastiano Vigna

Probabilmente poche persone tra quelle che stanno leggendo queste righe conosceranno TEX, e probabilmente un numero ancora minore sarà a conoscenza della persona e dell'opera di D. E. Knuth. Si rende quindi necessaria una piccola digressione.

D. E. Knuth è probabilmente il più importante studioso di informatica vivente. Chiunque abbia avuto a che fare con argomenti affini conoscerà la sua celebre opera *The Art of Computer Programming*, che raccoglie e discute centinaia di algoritmi di tutti i generi (numerici e non). Contrariamente però ad altri teorici, Knuth è anche un grande pratico, e una delle sue realizzazioni maggiori rimane senza dubbio il sistema TEX, descritto completamente in più libri pubblicati sotto il nome *Computers & Typesetting*; in particolare, i due testi *Metafont: The program* e *TEX: The program* definiscono, in un linguaggio derivato dal Pascal, una coppia di programmi destinati a lavorare sinergicamente alla produzione di testi stampati. Poiché le idee dietro a questi due programmi sono da una parte rivoluzionarie, ma dall'altra completamente controcorrente rispetto all'andamento attuale dell'industria informatica, mi è parso giusto dedicare loro un po' di spazio.

I Servizi di

Amiga Magazine offre una serie di servizi per agevolare i propri lettori nel reperimento di software e materiale utile alla programmazione. E' disponibile l'intera libreria di dischetti di pubblico dominio curata da Fred Fish. Ogni dischetto contiene numerosi programmi e utility, spesso corredati da listati sorgenti e commenti degli autori.
Per districarsi fra le centinaia di programmi disponibili nei dischi di Fred Fish, come in precedenza su Transactor per Amiga, pubblicheremo nei prossimi mesi l'elenco ragionato.
Tale elenco riporta, divisi per categoria, tutti i programmi presenti, completandoli con informazioni quali la descrizione della funzione, l'autore, il numero di versione, la disponibilità del sorgente e il disco nel quale sono contenuti. I dischetti possono essere ordinati contrassegnando i numeri desiderati, purché la quantità sia di almeno 3 dischi.

BUONO D'ORDINE

Completare il buono d'ordine (o una sua fotocopia) e spedire in busta chiusa a:
I servizi di Amiga Magazine, Via Pola 9 - 20124 Milano

Si può allegare: assegno, contanti o fotocopia della ricevuta di versamento **c/c n.11666203** intestato a **Gruppo Editoriale Jackson.** Non si effettuano spedizioni in contrassegno.

Desidero ricevere i seguenti articoli; contrassegnare con una X i numeri di Fish Disk desiderati (minimo 3)

❑ 1	❑ 12	❑ 23	❑ 34	❑ 45	❑ 56	❑ 67	❑ 78	❑ 89	❑ 100	❑ 111	❑ 122	❑ 133	❑ 144	❑ 155	❑ 167
❑ 2	❑ 13	❑ 24	❑ 35	❑ 46	❑ 57	❑ 68	❑ 79	❑ 90	❑ 101	❑ 112	❑ 123	❑ 134	❑ 145	❑ 156	❑ 168
❑ 3	❑ 14	❑ 25	❑ 36	❑ 47	❑ 58	❑ 69	❑ 80	❑ 91	❑ 102	❑ 113	❑ 124	❑ 135	❑ 146	❑ 157	❑ 169
❑ 4	❑ 15	❑ 26	❑ 37	❑ 48	❑ 59	❑ 70	❑ 81	❑ 92	❑ 103	❑ 114	❑ 125	❑ 136	❑ 147	❑ 158	❑ 170
❑ 5	❑ 16	❑ 27	❑ 38	❑ 49	❑ 60	❑ 71	❑ 82	❑ 93	❑ 104	❑ 115	❑ 126	❑ 137	❑ 148	❑ 159	❑ 171
❑ 6	❑ 17	❑ 28	❑ 39	❑ 50	❑ 61	❑ 72	❑ 83	❑ 94	❑ 105	❑ 116	❑ 127	❑ 138	❑ 149	❑ 160	❑ 172
❑ 7	❑ 18	❑ 29	❑ 40	❑ 51	❑ 62	❑ 73	❑ 84	❑ 95	❑ 106	❑ 117	❑ 128	❑ 139	❑ 150	❑ 161	
❑ 8	❑ 19	❑ 30	❑ 41	❑ 52	❑ 63	❑ 74	❑ 85	❑ 96	❑ 107	❑ 118	❑ 129	❑ 140	❑ 151	❑ 162	
❑ 9	❑ 20	❑ 31	❑ 42	❑ 53	❑ 64	❑ 75	❑ 86	❑ 97	❑ 108	❑ 119	❑ 130	❑ 141	❑ 152	❑ 163	
❑ 10	❑ 21	❑ 32	❑ 43	❑ 54	❑ 65	❑ 76	❑ 87	❑ 98	❑ 109	❑ 120	❑ 131	❑ 142	❑ 153	❑ 165	
❑ 11	❑ 22	❑ 33	❑ 44	❑ 55	❑ 66	❑ 77	❑ 88	❑ 99	❑ 110	❑ 121	❑ 132	❑ 143	❑ 154	❑ 166	

Nota: il n° 164 non è disponibile

❑ 3 Fish Disk L. 24.000 (abbonati L.18.000)

❑ L. 5.000 per ogni disco aggiuntivo riferito ad ogni singolo ordine

Tutti i prezzi sono da intendersi IVA inclusa e spese di spedizione comprese.

Firma ______________________________

(se minorenne quella di un genitore) Gli ordini non firmati non verranno evasi.

Cognome ______________________________

Nome ______________________________

Via ______________________________

Cap ____________ Città ______________________

Prov ____________ Telefono ______________________

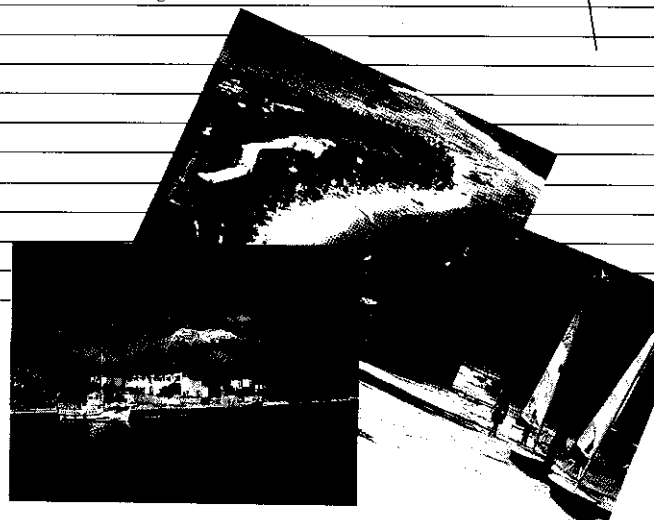
Codice abbonamento ______________________

ABBONAMENTI JACKSON: RISPARMI, VANTAGGI, REGALI

1990: anno di abbonamenti e privilegi assolutamente esclusivi, firmati Gruppo Editoriale Jackson. L'abbonamento anche ad una sola delle riviste Jackson vi assicura una lunga lista di privilegi: informazione e aggiornamento "recapitati" direttamente a casa vostra, senza alcuna spesa aggiuntiva, a prezzi superscontati, bloccati per un anno intero • la preziosa Jackson Card 90 che dà diritto a sconti nei numerosi esercizi convenzionati • un abbonamento gratuito alla rivista Jackson Preview Magazine e un supersconto a...sorpresa. Abbonarsi è facile: basta compilare e spedire la cedola che trovate in tutte le riviste Jackson.

TARIFFE ABBONAMENTO JACKSON '89-'90

RIVISTA	NUMERI ANNO	TARIFFA ABBONAMENTO
BIT	11	L.53.000 anziché L. 66.000
PC Magazine	11	L.52.000 anziché L. 66.000
PC Floppy	11	L.105.500 anziché L. 132.000
Informatica Oggi	11	L.52.500 anziché L. 66.000
Informatica Oggi Sett.	40	L.32.000 anziché L. 40.000
Trasmissione Dati e Tel.	11	L.51.500 anziché L. 66.000
Computergrafica & DTP	11	L.51.000 anziché L. 66.000
NTE Compuscuola	10	L.33.000 anziché L. 44.000
Elettronica Oggi	20	L.112.000 anziché L. 140.000
Automazione Oggi	20	L.80.000 anziché L. 100.000
EO News Settimanale	40	L.32.500 anziché L. 40.000
Strumentazione e Mis. Oggi	11	L.53.500 anziché L. 66.000
Meccanica Oggi	11	L.61.500 anziché L. 77.000
Strumenti Musicali	11	L.50.500 anziché L. 66.000
Watt	20	L.31.500 anziché L. 40.000
Fare Elettronica	12	L.58.000 anziché L. 72.000
Amiga Magazine (Disk)	11	L.123.500 anziché L. 154.000
Super Commodore (Disk)	11	L.110.000 anziché L. 137.500
Super Commodore (Tape)	11	L.75.000 anziché L. 93.500
PC Software 5 1/4	11	L.106.000 anziché L. 132.000
PC Software 3 1/2	11	L.132.000 anziché L. 165.000
PC Games 5 1/4	11	L.124.000 anziché L. 154.000
PC Games 3 1/2	11	L.132.500 anziché L. 165.000
Guida Videogiochi	11	L.31.000 anziché L. 38.500

...E 203 PREMI FAVOLOSI

E con il concorso abbonamenti 1990, il Gruppo Editoriale Jackson mette in palio premi straordinari: 3 viaggi con soggiorno per due persone in villaggi del Club Med' alle Mauritius, ai Caraibi, in Turchia e centinaia di praticissimi raccoglitori per floppy disk della MEE.

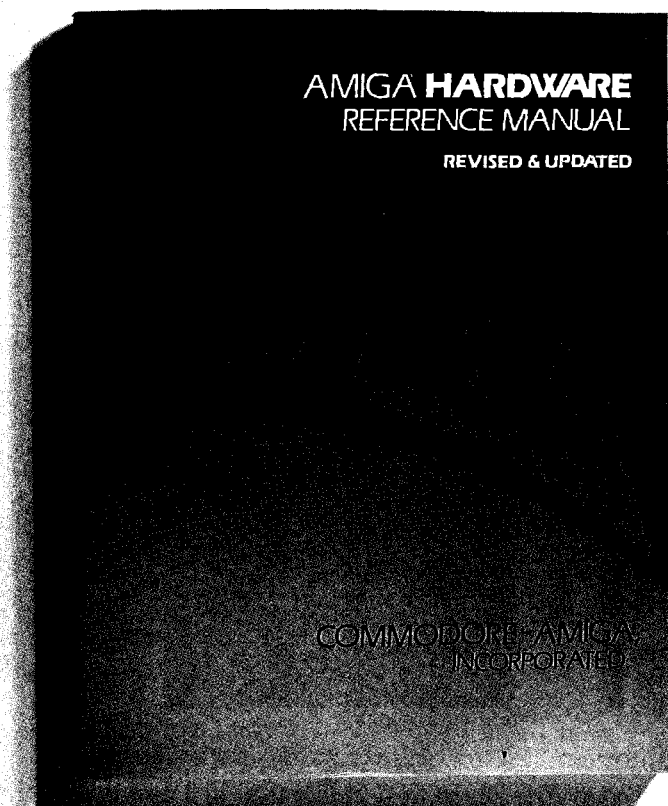
Conoscere l'hardware con la guida ufficiale Commodore

Stiamo parlando dell'Amiga Technical Reference Series, una collana specificatamente realizzata per i programmatori e gli sviluppatori, considerata come la loro "Bibbia".

Daniele Cassanelli

La serie Amiga Technical Reference costituì e costituisce l'unica documentazione ufficiale di Amiga per il programmatore. Curata dalla Commodore-Amiga Incorporated (tra gli autori spicca Rob Peck) e pubblicata dalla Addison Wesley, fece la sua prima comparsa in pubblico con il rilascio della versione 1.1 del sistema operativo e successivamente venne pubblicata per la release 1.2. Recentemente, dopo l'uscita del Kickstart 1.3, la Commodore ha commercializzato la suddetta serie aggiornata a questa versione, che, a differenza delle precedenti edizioni, è costituita da tre manuali che, oltre ad aver cambiato la copertina, riportano la dicitura "Revised & Updates" (riveduto e aggiornato). Il primo, Amiga ROM Kernel Reference Manual: Libraries and Devices, costituisce il punto di forza dell'intera collana, includendo una descrizione dettagliata di tutti i device e le librerie di Amiga, dall'Exec a Intuition (nelle precedenti edizioni, questo libro comprendeva tutti file include, mentre erano presenti due volumi separati appositamente riferiti all'Exec e ad Intuition). Il secondo libro della collana si intitola Amiga ROM Kernel Reference Manual: Includes and Autodocs e riporta tutti i file include (sia in C sia in Assembler) e la documentazione sul funzionamento di Amiga e sugli ultimi aggiornamenti al sistema operativo (FastFileSystem, Autoboot ecc.). Il terzo volume, Amiga Hardware Reference Manual, è quello di cui parliamo in questo articolo.

Amiga Hardware Reference Manual

Questo volume, in generale, intende spiegare il funzionamento di Amiga a livello hardware. Più specificatamente vuole aiutare i programmatori che utilizzano l'assembler a comunicare con il sistema saltando le routine delle library per rendere così il codice prodotto più corto e più veloce. Tutti i listati di esempio riportati dal libro sono, infatti, rigorosamente in assembler, l'unico mezzo per programmare Amiga a questo livello. Il libro (380 pagine) è costituito da 8 capitoli e 10 appendici. Il primo capitolo riporta una panoramica sull'hardware di Amiga e sul suo funzionamento generale.

Coprocessor

Già dal secondo capitolo si parla del COPPER e del suo ruolo nella macchina. Il COPPER è un coprocessore che risiede in uno dei chip custom di Amiga e ha il compito di controllare tutto il sistema grafico (ripristino degli sprite, cambio della palette, aggiornamento dei canali audio e controllo del blitter) e, grazie all'uso del DMA (Direct Memory Access), lascia libero il 68000, o chi per esso (68010, 68020 o 68030), di continuare l'esecuzione dei programmi. Nel capitolo vengono spiegate passo passo le istruzioni del COPPER, come funziona la copperlist e in che modo viene controllato il blitter.

Playfield & sprite

Il terzo e il quarto capitolo trattano della grafica di Amiga. Il capitolo 3 ha come tematica i playfield e inizia con una breve panoramica delle caratteristiche di uno screen, introduce definizioni di alcuni termini fondamentali e prosegue spiegando come si costruiscono e si gestiscono i playfield, siano essi normali o doppi (dual playfield) e come vengono organizzati i playfield in situazioni particolari, come lo scorrimento di un playfield sull'altro.
Il quarto capitolo tratta, invece, degli sprite: il modo con cui vengono definite grandezza, forma e colori, e come vengono posizionati e mossi su di un playfield.
Il capitolo passa poi alla trattazione di argomenti più complessi, ad esempio, su come ottenere più di 8 sprite contemporanei (limite teorico della macchina) sullo schermo. Su questo argomento è bene spendere due parole: questa tecnica utilizza i canali DMA riservati agli sprite; ognuno di questi può produrre, indipendentemente dagli altri, più di uno sprite alla volta rispettando alcune limitazioni. E' possibile che a volte si renda necessaria la visualizzazione contemporanea di più di 8 sprite, ad esempio per formare sprite più larghi (la larghezza massima di uno sprite è di 16 pixel), oppure per ottenere più di 4 colori sovrapponendo più sprite.
Per ottenere tutto ciò si utilizzano appunto i canali DMA creando precedentemente una struttura di dati che contenga tutte le informazioni necessarie allo scopo.
La sola restrizione da rispettare utilizzando questa tecnica è che gli sprite visualizzati tramite lo stesso canale DMA abbiano tra di loro almeno una distanza pari ad una linea video. Questo perché durante la fase di fetch (caricamento) in memoria della control word (doppio byte che descrive le caratteristiche dello sprite), non può essere visualizzato niente dal canale DMA che si sta utilizzando.

Audio

E' il tema del capitolo 5 che mostra come interagire direttamente con l'hardware che si occupa del suono. Inizialmente viene svolta una breve introduzione su come i computer possano produrre il suono per poi passare al caso specifico di Amiga, sulla programmazione dei quattro canali audio messi a disposizione dall'hardware della macchina.

Blitter

E' uno dei capitoli più interessanti dell'intero volume: il blitter è, infatti, uno dei due coprocessori di Amiga situato sul chip Agnus. Nel capitolo vengono mostrate tutte le applicazioni in cui il blitter svolge o può svolgere un ruolo fondamentale; vengono spiegate successivamente tutte le sue caratteristiche e la programmazione di questo velocissimo coprocessore. Alla fine del capitolo vengono riportati tre programmi in linguaggio assembly che utilizzano il blitter (memory clear, simple line e rotate bits).

System control

In questo capitolo (il settimo) vengono affrontati argomenti come la priorità dei playfield, la collisione degli oggetti grafici, il controllo del DMA e degli interrupt da parte del sistema e il reboot della macchina. Su quest'ultimo argomento viene fornito il codice sorgente per il reset di Amiga via software. Eccone una descrizione generale: dopo essere entrati in modo supervisore (tramite la routine dell'Exec Supervisor) viene caricato nel registro AO l'indirizzo 2, indicante la seconda word della ROM, che a sua volta contiene il codice dell'istruzione JMP che servirà per inizializzare il PC (Program Counter) del 68000. Successivamente viene eseguita l'istruzione RESET del 68000 (la quale può essere eseguita solo ed esclusivamente in modo supervisore), per poi saltare all'indirizzo contenuto in AO. In questo modo assemblando il tutto, otteniamo un comando che resetta la macchina, la cui lunghezza è di soli 64 byte.

Interface

Nell'ultimo capitolo vengono trattati argomenti inerenti alla comunicazione con il mondo esterno, vale a dire quindi gestione delle GamePort (mouse, trackball, joystick, penna ottica), della tastiera, del disk controller, della porta parallela Centronics, della porta seriale RS232-C e dei connettori video.

Appendici

Nelle dieci appendici, oltre a vari schemi di "cultura generale", vengono descritti i CIA (Complex Interface Adapters), meglio conosciuti con la sigla 8520; viene descritto il funzionamento dell'AUTOCONFIG e della tastiera, il "pinout" del connettore esterno dei dischi e viene riportata una mappa della memoria.

Conclusioni

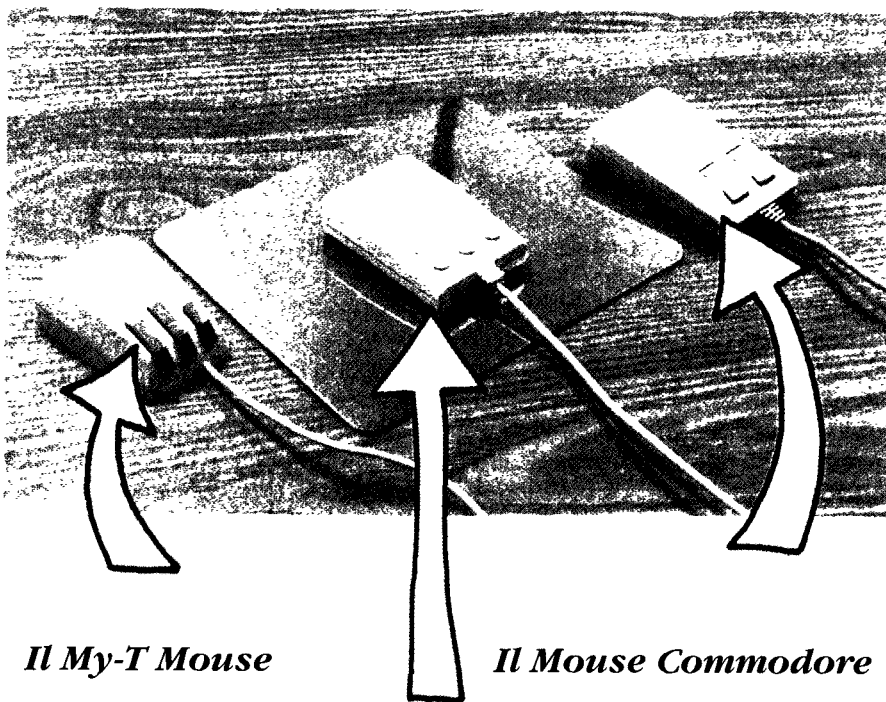
Il libro è sicuramente indirizzato a tutti i possessori di Amiga che intendono programmare questa stupenda macchina a livello hardware avendo così la completa gestione di Amiga in tutte le sue funzioni. Non possiamo che esprimere un giudizio più che positivo per una pubblicazione del genere visti, soprattutto, chi sono gli autori. ▲

Articolo tradotto dal n.3 Fall 1989 di Compute!'s Amiga Resource.

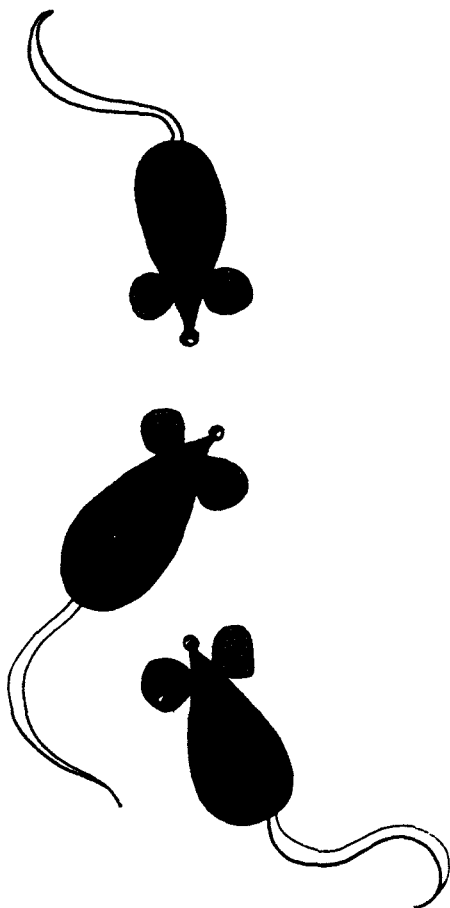
Uomini, topi e computer

Da quando è apparso alla ribalta, grazie a Macintosh, il mouse ha rappresentato lo standard dei dispositivi di input. E il "topo" è importante per Amiga esattamente come per Mac.

Rhett Anderson

Tutti i possessori di Amiga hanno un mouse collegato al loro computer preferito. Se ne può fare a meno, usando i tasti Amiga e i tasti cursore, ma non è esattamente divertente. Ora sono apparse due alternative al mouse: My-T Mouse, della Southern Technologies, e Boing! Mouse, della Centaur Software. In questo articolo esamineremo da vicino queste due novità assieme ad altri due dispositivi di input, la tavoletta Easyl e la Inkwell Light Pen.
Gli ultimi due prodotti citati si rivolgono più a chi ha interessi artistici o presunti tali, mentre per i mouse il discorso è diverso. Chi comprerebbe un mouse per Amiga, un computer venduto con mouse incorporato? Ci sono due possibili risposte: chi non ha più il mouse (per guasti, vecchiaia e simili cause) e chi vorrebbe sfruttare al meglio il software che supporta il terzo bottone del mouse, quello centrale, di cui sono dotati i nostri due modelli alternativi.
La seconda risposta oggi è poco realistica, visto che il software oggi in commercio difficilmente supporta il fatidico terzo bottone; tuttavia il software Amiga lo prevede, quindi

Il My-T Mouse

Il Mouse Commodore

Il Boing Mouse !

non meravigliatevi se prima o poi vedrete un fiorire di applicazioni che lo supportano.
Per quanto mi riguarda, trovo che la prima risposta possa essere valida: il mio primo mouse è morto dopo dieci mesi (subito dopo l'alimentatore), e il secondo, seppur funzionante, comincia a dare segni di stanchezza nonostante la mia attenta manutenzione periodica. Intendiamoci, non sto parlando male dei mouse Commodore, ma sono comunque lieto di vedere un po' di competizione sul mercato.
Mentre i mouse si collegano tutti alla porta apposita, le altre due periferiche, tavoletta e light pen, si connettono in altro modo, che varia secondo il software incluso nei rispettivi pacchetti.
La light pen si collega alla porta 2, quella del joystick, tranne che in Amiga 500, dove il dispositivo si connette alla porta di espansione, lasciando, comunque, un connettore identico a disposizione per espansioni di memoria, hard disk, eccetera.

Per un buon risultato, bisogna tenere la light pen a due centimetri dal video

Commodore Amiga Mouse
Incluso in tutti i modelli di Amiga
Commodore Italiana
Viale Fulvio Testi, 280
20126 Milano

Il mouse Commodore è naturalmente lo standard con cui ogni potenziale concorrente deve confrontarsi. La sensazione tattile è buona, e lo scorrimento della pallina è nella maggior parte dei casi fluido e silenzioso. Il cavo di collegamento, più lungo di un metro, si adatta virtualmente a qualsiasi configurazione di scrivania.
I problemi sono costituiti dal bisogno di pulizia regolare, pena l'insorgere di fastidiosi cigolii e la non perfetta affidabilità dei microinterruttori dei pulsanti, che tendono a fare di tanto in tanto cilecca.

The Boing ! Mouse
Centaur Software
Distribuito da Creative Computers
4435 Redondo Beach Blvd.
Lawndale, CA 90260

Boing! è il primo mouse ottico per Amiga. Subito da elogiare per il suo eccellente feeling, la silenziosità e la morbidezza del movimento. A differenza dei mouse meccanici, Boing! non ha parti in movimento, e quindi non abbisogna di manutenzione.
La tecnologia su cui si basa un mouse ottico come Boing! prevede il monitoraggio di un raggio di luce proveniente dal corpo del mouse. Sfortunatamente, unico punto a sfavore, il grado di accuratezza non è alto come quello dei modelli tradizionali, e il puntatore crea problemi, specie quando viene mosso molto velocemente attraverso lo schermo.
Boing! è dotato di tre bottoni. Attualmente quello centrale è pressoché inutilizzabile, per la mancanza di software che lo sfrutti, ma la situazione sta già cambiando in meglio. Prova ne siano i programmi con questa opzione in via di apparizione, primo fra tutti l'interfaccia X-Windows sviluppata da GfxBase (società fondata da Dale Luck, autore di buona parte della libreria grafica di Amiga) per usare Unix su Amiga.
Se siete disposti alla spesa di un mouse supplementare e non volete preoccuparvi della manutenzione del vostro mouse, sarete piacevolmente impressionati da Boing!.

The My-T Mouse
Southern Technologies
2009 McKenzie
Suite 110
Dallas, Texas 75006

Questo mouse, sebbene abbia un'aria un po' troppo economica, è un ottimo sostituto del mouse di Amiga, soprattutto per la sua accuratezza. Durante l'uso, The My-T Mouse emette un suono non dissimile da quello del mare in una conchiglia. Qualcuno troverà la cosa sgradevole, ma a me non ha dato disturbo.
Come Boing!, anche My-T Mouse dispone di tre bottoni, forse un po' piccoli per la mano di un adulto; il feeling complessivo è però soddisfacente, e My-T Mouse merita tutto

sommato di essere consigliato.

Inkwell Light Pen
Inkwell Systems
1050-R Pioneer Way
El Cajon, CA 92020

Devo confessare che il primo impatto con la Inkwell Light Pen è stato piuttosto frustrante. Inizialmente, la penna non vedeva bene il mio schermo, poiché uso uno sfondo nero. Siccome la penna ottica riconosce la luce emessa dal tubo catodico del monitor, deve arrivarle un segnale abbastanza forte da essere identificato. E nel gergo dei tubi catodici, forte significa luminoso, al punto che potreste persino dover lavorare con occhiali da sole per evitare fastidi.
Dopo avere caricato DeluxePaint III (con la versione II la penna non funziona), ho scelto uno sfondo di colore bianco, e la penna funzionava perfettamente.
Devo dire che ho fatto una certa fatica ad adattarmi all'uso della Inkwell Light Pen, e che non posso dire, probabilmente a differenza di altri, di esserne diventato utente provetto.
La cosa più difficile è stata capire dove posizionare il gomito. Ho dovuto spostare il computer sulla sinistra per lasciare spazio libero davanti allo schermo, ma anche così il braccio si stancava dopo pochi minuti.
La Light Pen non è accurata se la premete contro lo schermo. La distanza migliore a cui mantenerla si aggira sui cinque centimetri.
Il prodotto potrebbe essere utile come dispositivo di input per programmi personalizzati. Per esempio, si potrebbe scrivere un programma che consenta di effettuare con la penna gli inserimenti previsti dai menu.
Dotata di un buon manuale e di sufficiente software a corredo (che rende possibile la coesistenza tra penna e mouse), la Inkwell Light Pen, per essere una penna ottica, funziona bene, ma francamente non vedo i vantaggi che dovrebbero indurmi a rinunciare al mouse.

Con Easyl gli artisti possono disegnare nel loro modo abituale: con carta e penna

Easyl
Quest
Via E. Da Prisco, 24
37136 Verona

Agli artisti la Easyl (già recensita sul numero 9 di Amiga Magazine) piacerà molto, specialmente quelli abituati a disegnare su carta. E' molto più naturale disegnare su una tavoletta, che non farlo muovendo il mouse. Per usare la Easyl, basta appoggiare un foglio di carta sulla tavoletta, far partire il vostro programma di grafica preferito e cominciare a disegnare con una matita o una penna.
L'installazione della Easyl avviene inserendo la sua interfaccia nel connettore di espansione di Amiga. Sul nostro Amiga 500 l'operazione risultò più difficile del previsto; l'interfaccia entrava a fatica e, una volta completato l'inserimento, il boot non avveniva, lasciandoci con uno schermo verde e una spia di accensione intermittente.
Fortunatamente, il nostro collega Randy Thompson aveva letto da qualche parte che è possibile resettare alcuni chip dentro il computer, schiacciando quest'ultimo su un ginocchio. Non ci credo ancora, ma ha funzionato.
Easyl è completa di un programma omonimo, a cui però preferiamo di gran lunga DeluxePaint III. Il mouse non viene disturbato dalla presenza della tavoletta.
Il software del driver funziona sia per i destrorsi che per i mancini. I driver normali usano i bottoni del mouse nel modo consueto, mentre quelli preceduti dalla lettera R (reverse, inversione) invertono il funzionamento del bottone di sinistra.
L'effetto risultante è che non è più necessario tenere premuto il bottone mentre si disegna, ma solo se si desidera spostare la penna sul documento.
Per chi si sente artista nell'animo e trova difficoltà a ottenere gli effetti desiderati con il mouse, la Easyl è una scelta eccellente. ▲

Scopri i segreti di AMIGA

Backup, dischi e altro

Sebastiano Vigna

Vi è mai capitato di perdere seriamente dei file? Non l'ultimo gioco che un amico vi ha "prestato", intendo roba vostra. Molti utenti pensano che il backup del disco rigido sia una attività inutile. Quello che vi voglio dimostrare è che non è neppure sufficiente.

Qualche giorno fa armeggiavo col mio HardFrame, di cui ho tanto parlato la volta scorsa. Essendomi accorto di aver dato un po' troppo spazio ai file e un po' poco ai programmi, avevo deciso di cambiare partizioni, da 15/25 a 10/30. La cosa sembrava, piuttosto facile, tanto più che mi ero procurato il nuovo programma di configurazione della Microbotics, che sebbene ancora allo stadio beta (leggi: qualche buco da tappare) disponeva di una magnifica interfaccia grafica.

Premetto che negli ultimi tempi il mio secondo floppy faceva un po' le bizze. In particolare, quando accendevo Amiga cominciava ad emettere un click molto più sonoro del normale, e si rifiutava di leggere un disco fino a che quest'ultimo non era stato reinserito 2-3 volte, dopodiché funzionava perfettamente.

La fase uno è stata ovviamente fare un mega-backup di tutt'e due le partizioni. Il programma che uso, scrive direttamente le tracce del disco e le verifica, bypassando AmigaDOS. Dopo aver ripartizionato e formattato l'hard disk ho cominciato a riversare di nuovo il tutto. E qui sono cominciati i problemi.

La prima partizione è tornata al suo posto, ma durante il restore dell'altra, il programma si è rifiutato di leggere il secondo dischetto. Una breve indagine mi ha confermato che quest'ultimo era completamente illeggibile, tutte le tracce erano errate. Nel caso non aveste capito, avevo inserito il verify, che quindi si è rivelato essere molto affidabile.

La prima reazione è stata ottimistica: va bene, in fondo sul disco rigido ci sono solo sei file veramente preziosi di cui non hai alcuna copia, ovvero i sorgenti dell'IFF viewer universale a cui stai lavorando. Figurati se sono proprio andati a finire lì dentro...

Ovviamente lo erano.

Alla riscossa!

Dopo alcuni minuti di silenzioso dolore, mi sono tirato su e ho pensato lungamente. Dato che le partizioni vengono riempite in modo da minimizzare il movimento della testina, i file sono concentrati in mezzo, e dunque i dati che avevo ricaricato dovevano occupare un'area solo in parte coincidente con i vecchi file. Poiché la formattazione era stata QUICK, solo il root block era andato perso, i file potevano essere ancora lì. Detto, fatto: ho preso la mia copia di DiskSalv, versione 1.42, ho rimesso a posto le partizioni esattamente come prima e l'ho lanciato sul quella dei file, che a questo punto conteneva un magma inestricabile di dati, con doppioni e false indicazioni. I miei complimenti a Dave Haynie per questo meraviglioso programma, che mi ha salvato la vita: i sorgenti erano ancora là, e dopo circa un'ora di tempo e sei dischi di spazzatura sono riuscito a recuperarli. Qualcos'altro è andato perduto, i sorgenti C per usare ARP, ma con qualche ora di lavoro penso di poterli riprodurre.

Morale: una copia di backup non basta. Se ci pensate, su un disco ci sono veramente poche cose a cui tenete sul serio. I dati che, se persi, possono veramente mandarvi in crisi sono meno dell'1%, in media. Ad esempio, un articolo si recupera in qualche ora, ma ci sarebbero volute settimane per riscrivere tutti i sorgenti C... per non parlare del debug! Meglio tenerne tre o quattro copie, possibilmente in luoghi diversi. Non rompete la testa contro il muro dopo aver versato una lattina di Coca-Cola sulla scatola dei dischetti ove tenevate insieme sette copie del vostro lavoro, pensateci prima !

Non è comunque semplice scegliere un programma di backup, e sull'argomento esistono molte diverse posizioni. Una prima divisione va fatta tra i programmi che copiano i file in formato AmigaDOS su dischetti, come il DoubleBack contenuto nel Disk Mechanic della Lake Forest Logic, e quelli che accedono direttamente alla traskdisk.device e scrivono i loro dati traccia per traccia, come il QuarterBack, eventualmente comprimendoli. I primi sono molto lenti, ma certamente più sicuri, in quanto non si può incappare facilmente in dischi disallineati, e, ad esempio, in fase di backup mi sarei accorto di quello che stava succedendo. D'altra parte, i secondi occupano meno spazio, sono molto più veloci e, a meno di casi estremi come il mio, danno una notevole sicurezza. Un problema che può però insorgere è quello del carico o scarico di un singolo file o di un gruppo.

Tutti i programmi di backup a traccia da me visionati permettevano di caricare dai floppy un gruppo di file selezionati, ma non permettevano di

farne il backup, quindi o dividete la vostra partizione in blocchi differenziati o dovrete fare ogni volta il backup di tutto.

Software negli USA

L'ultima volta ho raccontato delle traversie in cui sono incappato cercando di procurarmi hardware e software negli USA. Sembra però che questo problema da oggi non esista più. Creative Computers, la ditta di cui avevo segnalato il numero verde, ha aperto una filiale in Germania che dovrebbe vendere agli stessi prezzi la stessa varietà di prodotti sul mercato europeo. Se la cosa ha successo, dovrebbe essere una enorme facilitazione, data la distanza molto minore e la differenza di trattamento doganale. Inoltre, questo fatto dovrebbe spronare le ditte per corrispondenza italiane ad abbassare i prezzi. Il numero è (49) 40-407332, e non avendo io nessun'altra informazione non c'è che da provare. Tra parentesi, il livello minimo di spesa per poter usare il numero verde negli USA è stato portato a $200.
La Soft-Logik, che produce PageStream, mi ha mandato una "gentile" lettera in cui afferma che non riesce a star dietro alla posta. A causa di questo fatto, mi rimanda la lettera di bug report sperando che la release 1.8 del prodotto risolva i miei problemi (problemi? MIEI?). Evidentemente, hanno ricevuto così tante segnalazioni di buchi da non poterci stare dietro, e hanno preferito rimandare il tutto. Quel che però mi rende furioso è che da amici americani (i quali hanno già ricevuto la 1.8) ho saputo che almeno uno dei buchi da me segnalati è ancora lì. La versione sembra estremamente più stabile della 1.6, e anche se qualcuno l'ha malignamente definita "almeno ad uno stadio da beta-test", è sempre meglio che niente. Sempre meglio, ad esempio, della Taurus-Impex, che a sua volta ha mandato ad un mio amico una (sempre gentile) lettera per avvisarlo che hanno chiuso il loro reparto di sviluppo per Amiga, e che mentre X-Cad sarà supportato dalla CadVision, Acquisition, il database, viene semplicemente "mollato", il che equivale a dire che un investimento di 300.000 lire va a farsi friggere. Bene, ci vogliono proprio software house come queste per mandare avanti la causa Amiga...
Sul versante positivo, Bill Hawes mi ha appena mandato la nuova release di WShell, certamente la più potente shell per Amiga (forse rispetto alla 1.05 la shell di ARP aveva parecchio da dire, ma ora...). Oltre al solito giro di bug fixing, ci sono parecchi piccoli miglioramenti e, soprattutto, una nuova utility "esterna", FComp, che rappresenta lo stato dell'arte in fatto di file completion. Vediamo un po' di spiegare di cosa si tratta.
Supponiamo che voi dobbiate copiare il file devs:system-configuration su un altro disco, e non vogliate usare i wildcard (per esempio perché c'è un file DEVS:system-qualcos'altro). Dovreste scrivere il nome per intero, cosa certamente non piacevole. Qui entra in gioco la file completion: voi scrivete "DEVS:sy", premete ESC e FComp vi scrive il nome completo del file. Se ci sono più possibilità, potete "aiutarlo" inserendo altri caratteri, oppure premere ancora ESC. Sto dando una visione molto semplificata: in realtà ci sono decine di opzioni che permettono un aumento della vostra performance alla CLI di un ordine di grandezza. E ovviamente, FComp ha un algoritmo molto efficiente di cache per le directory, altrimenti le attese sarebbero intollerabili.
Certe volte mi chiedo come fa Bill ad avere sempre le idee giuste. Voglio dire che una volta che ci si è prefissi di scrivere un certo programma, prima o poi ci si riesce: il vero genio sta nel trovare dei programmi intelligenti da scrivere.
Una situazione analoga si trova in matematica per i teoremi, ed in effetti, quando ad un teorema è associato un nome, nel 90% dei casi la dimostrazione che viene insegnata è di qualcun'altro, successivo, non quella originale; in un certo senso quel che importa è l'enunciato. Altro esempio di programma intelligente è PathMan, anch'esso incluso in WShell, che permette di "spargere" una periferica logica (as esempio DEVS:) su più directory fisiche. Allo stato attuale, AmigaDOS non permette assegnazioni multiple di questo tipo, e PathMan risulta molto utile sia ai possessori di floppy che non vogliono riassegnare tutto ogni volta che devono lanciare un programma che ha delle librerie o delle periferiche speciali, sia ai possessori di hard disk che non vogliono concentrare tutti i font o tutte le librerie in un'unica directory. PathMan agisce trasparentemente rispetto ad AmigaDOS, e se, ad esempio, abbiamo dato "Assign DEVS: PATH: df0:devs, df1:devs", quando un programma cercherà di aprire una periferica (ovvero un file che termina con ".device"), PathMan gli passerà prima la lista di quelle disponibili in df0:devs, poi la lista di quelle disponibili in df1:devs; di tutto questo il programma-cliente è ovviamente ignaro. Non ho mai riscontrato gravi problemi di compatibilità e l'unico pericolo era costituito da alcuni file requester che, incontrando i filename di PathMan (possono essere molto lunghi), andavano in crash. Nella 1.2 Bill ha risolto, direi brillantemente, la situazione permettendo di applicare un "commento" ai path mediante il comando standard "FileNote". Se, ad esempio, nella startup-sequence noi avevamo dato Assign FONTS: PATH:df0:, df1:, df1:fonts, TeX:fonts, RAD:, RAM: tutta la parte dopo PATH: sarebbe apparsa ad AmigaDOS come un filename. Essendo più lunga di 30 caratteri, si potevano avere grossi problemi (leggi crash indesiderati) quando si usavano file requester non molto intelligenti. La soluzione di Bill è semplicissima: il comando FileNote FONTS: <nome> fa sì che <nome> diventi il nome del file "finto" che AmigaDOS vede, mentre la lista di directory diventa un commento. Basta quindi dare <nome> con meno di trenta caratteri per risolvere ogni problema.
Se solo tutti i pacchetti per Amiga girassero così bene... ▲

Come utilizzare il dischetto

Prima di utilizzare il dischetto allegato alla rivista assicuratevi di averlo protetto dalla scrittura aprendone la finestrella, per evitare accidentali cancellazioni o contaminazioni da virus (il disco allegato è al 100% privo di virus ed il Gruppo Editoriale Jackson non si assume nessuna responsabilità in caso di contagio). A questo punto accendete il computer. Dopo la presentazione, dalla quale è possibile uscire premendo il tasto sinistro del mouse, verrà caricato il Workbench. Aprendo l'icona del disco (cliccando due volte con il tasto sinistro su di esso) appaiono i "cassetti" relativi a giochi, utilities, ecc. Aprite allo stesso modo il "cassetto" al quale siete interessati e caricate il programma desiderato. Alcuni programmi possono essere anche caricato da CLI con altre opzioni (vedere a parte le spiegazioni dei programmi). Per chi possiede Amiga con "soli" 512 K di memoria RAM è consigliabile chiudere tutte le finestre aperte tranne quella relativa al programma da far girare (che può essere eventualmente chiusa con un po' di prontezza di riflessi durante il caricamento). In tal modo il programma avrà a disposizione più memoria. In caso contrario, potrebbero verificarsi dei malfunzionamenti.
ATTENZIONE: prima di far partire ogni programma vi raccomandiamo di leggere attentamente le istruzioni del programma.
I programmi Access e Mackie sono di pubblico dominio , e possono essere liberamente distribuiti. Inoltre, ricordiamo che, i programmi sorgenti ed eseguibili delle pagine di TRANSACTOR per AMIGA, si trovano nel cassetto TRANSACTOR e devono essere caricati da CLI. Tutti gli altri programmi presenti sul dischetto sono, invece, coperti da Copyright 1990 Gruppo Editoriale Jackson-Compute! Publications Inc., e come tali non devono essere copiati e distribuiti senza la preventiva autorizzazione scritta dell'editore. Amiga Workbench 1.3 è copyright 1985, 1986,1987,1988 Commodore-Amiga Inc., tutti i diritti sono riservati.

Per ogni questione tecnica riguardante il funzionamento del disco telefonare il Martedì e il Giovedì dalle ore 14.30 alle ore 16.00 al 02-6948287.

Come entrare in CLI

- Accendere il computer
- Inserire il disco con il Workbench
- Attendere il caricamento
- Cliccare due volte sul disco del Workbench
- Cliccare due volte sull'icona SHELL (per il nuovo CLI), oppure aprire il il cassetto System
- Selezionare l'icona CLI

Esempi di caricamento da CLI
- Entrare in CLI
- Digitare: COPY FROM C: TO RAM:
- Inserire il disco di Amiga Magazine e digitare CD DF0:
- Digitare CD UTILITIES

A questo punto, per esempio, si può digitare:

- Mackie
- Access

Amiga Magazine n. 12

SERVIZIO LETTORI **Compilare e spedire in busta chiusa a: GRUPPO EDITORIALE JACKSON Area Consumer - Via Pola, 9 - 20124 Milano**

A) Come giudichi questo numero di Amiga Magazine ?

❑ Ottimo
❑ Molto Buono
❑ Buono
❑ Discreto
❑ Sufficiente
❑ Insufficiente

B) Quale (i) articolo (i) o rubrica hai apprezzato di più?

Quale meno?

C) Cosa ti piacerebbe leggere nei prossimi numeri di Amiga Magazine ?

D) Ti è piaciuto il Software ?

E) Quante persone leggono la tua copia di Amiga Magazine ?

F) Possiedi un computer? Quale?

Quale (i) computer intendi acquistare in futuro?

G) Leggi altre riviste Jackson?
❑ SI ❑ NO

Quali?

H) Leggi altre riviste del settore?
❑ SI ❑ NO

Quali?

I) Oltre alle riviste dedicate al computer quali sono le tue letture preferite?

L) Quali sono i tuoi hobbies e maggiori interessi?

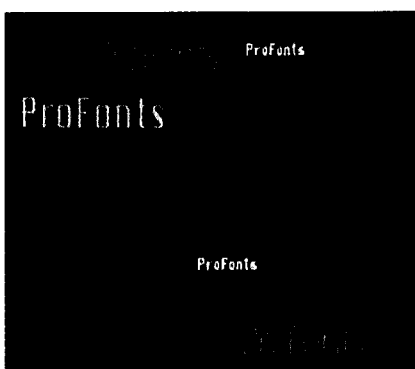
❑ Sport
❑ Musica
❑ Videoregistrazione
❑ Hi - Fi
❑ Fotografia
❑ Automobile
❑ Moto
❑ Viaggi

Nome
Cognome
Indirizzo
Età **Professione**
Città
Prov. **C.a.p.** **Tel.**

Tre programmi per tutte le occasioni

Rhett Anderson

ProFonts
New Horizon Software

ProFonts Volume I: Professional e ProFonts Volume II: Decorative sono complementi al nuovo word processor ProWrite della Horizon. I dischi contengono numerosi font bitmap per Amiga, in molte dimensioni o, meglio, corpi.
Mentre il volume I comprende font di alta qualità per il lavoro "serio", il volume II raccoglie caratteri più orientati artisticamente, utili per titoli e messaggi particolari.
Essendo font standard Amiga e non PostScript, funzionano bene anche nei programmi grafici e in altri word processor.
La documentazione è accurata, arrivando a fornire fogli che mostrano l'aspetto su carta di ogni font. Questo è abbastanza importante, quando si lavora con set di caratteri chiamati Gold, Silver, Quartz e così via.
In ogni dischetto è compreso System Mover, una utility che permette di installare e rimuovere i font con grande facilità.
Consigliamo a tutti di tenerne una copia dentro la cartella di sistema.

Rock Challenge
ReadySoft

Rock Challenge è un classico gioco a quiz, del genere "trivia", dedicato al rock'n'roll.
Il gioco ricorda vagamente il vecchio Rischiatutto televisivo, in cui le domande venivano raggruppate in categorie che in questo gioco sono Award (riconoscimenti, premi), Album (LP), People, Song e Date (date e ricorrenze).
Giocabile da 1-4 persone, Rock Challenge è divertente, ma a patto di essere almeno in due.
Lo schema di gioco comprende domande dal valore di 100, 200 e 300 punti, più una domanda jolly con punteggio a sorpresa.
Il gioco è un'ottima occasione per ripassare la propria cultura rock dagli anni '50 agli '80, ma non tutto funziona come dovrebbe. Intanto le domande da 300 punti non sembrano essere più difficili delle altre, e a volte le risposte sono piuttosto ambigue. In più, dopo poche partite, cominciano a comparire domande già viste, che rendono il gioco noioso. Le istruzioni non specificano il numero di domande comprese, approssimabile però dal fatto che il floppy è pieno al 75% circa.
In ogni caso, Rock Challenge resta un gioco divertente e consigliabile.

CygnusEd Professional
ASDG

Non vorrei usarlo per scrivere una lettera; e, infatti, CygnusEd Professional non è nato per essere un word processor; è uno strumento per programmatori che hanno bisogno di editare codice sorgente velocemente e senza problemi. E' stato scritto per essere il text editor più veloce e versatile tra quelli per Amiga, e lo è davvero.
Il manuale parla di una velocità di scroll dello schermo di 30 mila caratteri per secondo. Non so se sia vero, ma sicuramente non sono mai riuscito a muovermi attraverso un documento alla velocità che ho riscontrato con questo programma.
Oltre a fornire pieno supporto del mouse e menu a discesa, CygnusEd Professional supporta macro da tastiera personalizzabili (per emulare il comportamento del proprio editor preferito), operazioni "hot key" (l'editor rimane residente in memoria, attendendo la pressione di un tasto determinato), editing multiplo, interfaccia ARexx, selezione di blocchi rettangolari di testo, recupero di file non salvati dopo una Guru Meditation e numerose altre opzioni.
Sono presenti anche il word wrap e la stampa dei file, anche se un po' macchinosa (e questa è una delle ragioni per cui non lo userei come word processor).
Problemi? La memoria. Il programma occupa una porzione considerevole di spazio su disco e in RAM rispetto ad altri text editor. Ma se non si hanno di queste preoccupazioni, CygnusEd Professional è il text editor oggi più efficiente e utile disponibile. Non vorrei mettermi a scrivere un programma senza di lui. ▲

Audio Video Digitizer

Questo mese abbiamo provato, per la gioia di tutti i nostri lettori, una periferica che occupa una posizione di tutto rispetto nel mondo Amiga: l'Audio/Video Digitizer.

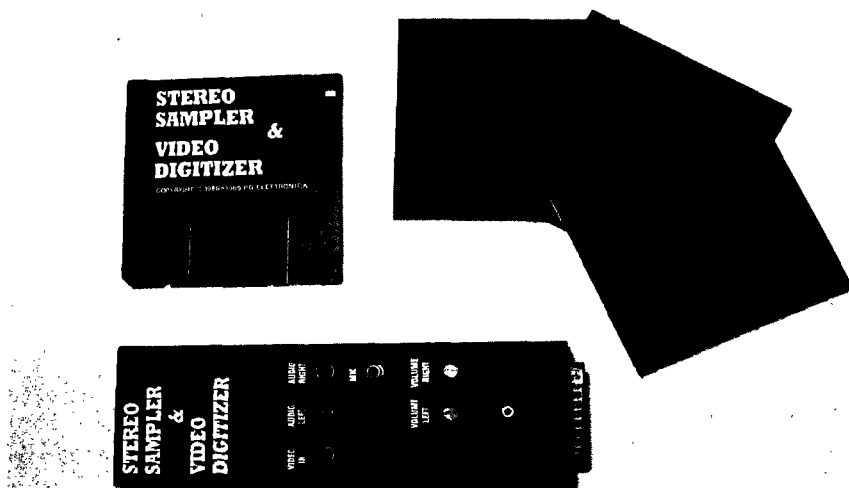
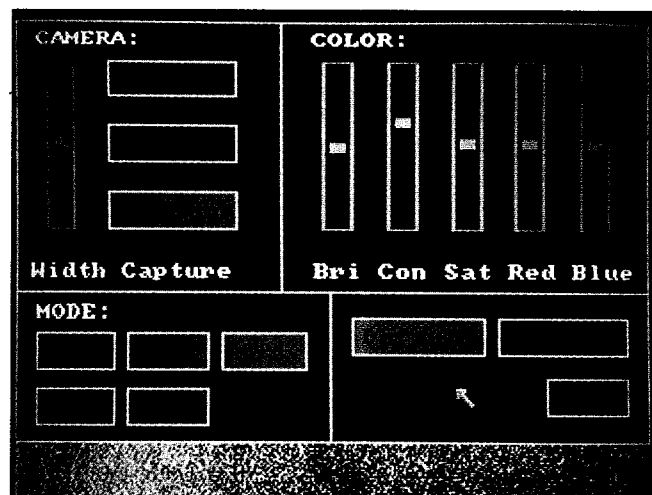
Stefano Paganini

La semplicità di Amiga (user-friendly) e del suo software, oltre alla potenza della macchina, non poteva sfuggire al settore video e, come noto, anche in alcune trasmissioni della RAI vi sono animazioni genlockate generate da Amiga.
I vantaggi che una soluzione basata su Amiga può offire all'utente professionale, una volta considerate le ovvie limitazioni della macchina, sono molteplici e le alternative sono spesso accessibili solo a costi molto elevati. In altri casi Amiga può fungere da complemento del sistema già in uso e permettere di realizzare in tempo reale anche animazioni di prova quali basi per la versione definitiva sul sistema a maggior risoluzione grafica e cromatica.
In ogni caso si tratta della stazione desktop video più interessante per il rapporto prestazioni/prezzo.

I Digitalizzatori Audio

Per quanto concerne i digitalizzatori audio, Amiga dispone di una sezione sonora che permette campionamenti fino a 28.8 KHz e l'impiego dei suoni campionati con strumenti MIDI pone Amiga di fronte ad una concorrenza agguerrita sia da parte di ATARI ST sia dai più professionali sequencer che, con l'aggiunta di hard disk da 80 Mb e più, permettono di raggiungere i 44.1 KHz del CD.
Anche in questo caso le considerazioni che riguardano il mercato amatoriale sono più che a favore di Amiga, grazie anche a parecchi pacchetti di editing musicale e dei suoni campionati, come, ad esempio, Aegis AudioMaster II.

La Prova

Il digitalizzatore in prova questo mese è prodotto dalla torinese P.G. Elettronica e comprende in un unico package sia il campionatore audio stereo sia il digitalizzatore video.
Si tratta di un abbinamento classico in quanto entrambe le sezioni sono compatibili con i rispettivi standard

di riferimento: il digitizer Digi-View e i sampler FutureSound.
Il digitizer vero e proprio è uno "scatolotto" di dimensioni ridotte, ma dal peso non indifferente tanto che sulla confezione viene indicato solo per Amiga 500.
Il motivo sta nel fatto che il collegamento con la porta parallela sul 500 consente di porre il digitizer allo stesso livello del computer mentre per il 2000, come indicato nella documentazione acclusa, è necessario un supporto, mentre per Amiga 1000 è necessario un doppio connettore femmina-femmina, ricordando tuttavia che dal modello 1000 ai 500/2000 la parallela ha subito delle variazioni anche nella disposizione dei pin (voltaggio non presente etc.). Sul dorso dell'unità sono presenti i jack RCA per i collegamenti Video e Audio Left e Right oltre all'ingresso microfono monofonico con un jack da 3,5 mm.; vi sono, inoltre, due trimmer di regolazione per il volume dei due canali.
Insieme al digitizer vengono forniti i tre filtri (rosso, verde e blu), un manualetto di istruzioni e il dischetto con i relativi programmi.
L'installazione, rigorosamente a computer spento, è molto semplice, e prevede il solo inserimento del digitizer nella parallela oltre ai cavetti in ingresso da telecamera, VCR o altra sorgente video, oppure un segnale audio in stereo oppure mono. Riteniamo quasi superfluo sottolineare la praticità di una doppia soluzione che evita di dover spegnere Amiga per collegare il modulo audio al posto del video e viceversa, come accadrebbe con altri modelli.
Il caricamento del software di gestione avviene nel modo consueto; il disco contiene un programma SAMPLER e tre versioni per la digitalizzazione video, rispettivamente in bassa risoluzione 320 X 200, per Amiga con 512 k di RAM, e la versione LO-RES da 1 Mb con risoluzione PAL 320 X 256; l'ultima versione funziona a 640 X 256 con 16 livelli di grigio. Per quanto riguarda i due programmi LO-RES, funzionalmente simili, consentono la selezione della modalità cromatica, da 16 colori al modo HAM+, cioè HAM 4096 colori ottimizzati da un algoritmo che tiene conto della ricorrenza di 16 colori base. I menu sono suddivisi in: PROJECT, con le consuete opzioni di Load/Save di file IFF o RAW, DIGITIZE e COLOR.
Il menu DIGITIZE, contiene quattro opzioni, le più usate, attivabili anche da tastiera, e che digitalizzano ciascuna uno dei tre colori fondamentali nella sintesi cromatica additiva: rosso, verde e blu.
Ad ogni passaggio sarà necessario cambiare i filtri sull'obiettivo della telecamera, l'operazione può essere automatizzata in parte grazie all'opzione Automatic che effettua i tre "passaggi" con un intervallo di qualche secondo per permettere il cambio di filtro, seguendo rigorosamente la sequenza rosso-verde-blu. Il menu COLOR contiene tutte le opzioni per modificare i parametri di digitalizzazione e di visualizzazione: da segnalare la possibilità di variare la velocità di digitizing da 7 secondi, per prove, ai 30 secondi del modo MEDIAN, che calcola una media tra due differenti campionamenti. La versione HI-RES differisce principalmente nel menu DIGITIZER, nel quale compaiono solo due scelte, differenti nella qualità dell'immagine in output, ma sempre in 16 livelli di grigio. Anche il pannello CONTROL è mutato di conseguenza, ed è presente il gadget SHARPNESS, per modificare la "grana" dell'immagine. Su entrambe le versioni è disponibile la funzione HISTOGRAM, che mostra i valori per R-G-B dell'immagine originale e di quella modificata, tramite CONTROL.
I risultati migliori arrivano con un po' di pratica anche se vi sono dei parametri che andrebbero sempre tenuti presente: il tempo di "esposizione", maggiore è, più accurato sarà il campionamento, almeno per la versione definitiva dell'immagine, come ad esempio il modo MEDIAN.
Le migliorie più significative si possono ottenere tramite i comandi di controllo della saturazione e contrasto e, in particolare, il controllo SHARPNESS nella versione ad alta risoluzione. Una delle più gravi carenze per un programma Amiga è la mancanza della funzione UNDO,

che in questo caso non è stata implementata neppure nella versione per memoria espansa.
Stupisce l'assenza delle modalità interlacciate peraltro presenti nella maggior parte degli altri digitalizzatori presenti sul mercato e sullo stesso Easy View, la versione "solo Video" che la P.G. Elettronica propone nella gamma dei suoi prodotti.

Audio

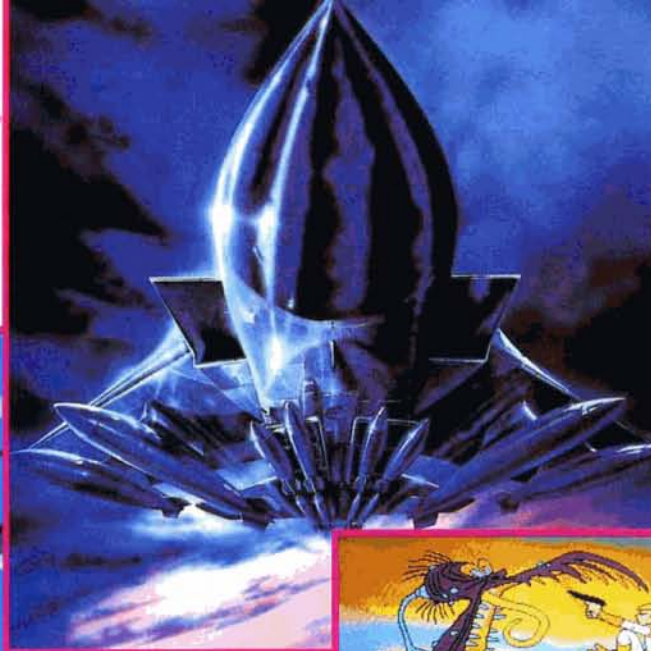
Passiamo ora alla sezione AUDIO e al programma SAMPLER, che indipendentemente dalla quantità di memoria disponibile, lascia sempre circa 350-400 Kbyte disponibili per il chip sonoro. Il funzionamento è molto semplice e, la disposizione dei gadget sullo schermo (nessun menu) è essenziale.
La finestra centrale contiene una sorta di grafico del campione in memoria. I parametri in questo caso riguardano il tempo di campionamento e la relativa frequenza, fino a 20 KHz, la selezione e la regolazione del volume dei due canali, particolare curioso in quanto la maggior parte dei campionatori arriva fino al limite massimo di 28 KHz!
Il gadget Hi-Fi disabilita il filtro presente all'interno di Amiga e migliora l'output, con il caratteristico calo di luminosità del LED di accensione.
Sulla sinistra, invece, vi sono i gadget di controllo, Load/Save, Record/Play e le due interessanti funzioni EDIT e Monitor. Edit consente di effettuare un set di operazioni sul suono campionato che comprendono, tra l'altro, il MIXing delle due tracce presenti in memoria e l'effetto FLIP, che equivale ad un play in reverse della traccia. Per inciso il manuale stesso fa riferimento al programma leader in questo senso: AudioMaster II, che risulta pienamente compatibile con il modulo digitizer e che permette di sfruttarne appieno le potenzialità. SAMPLER consente comunque, il salvataggio di file IFF audio compatibili con qualunque altro programma Amiga.
Nel modo Monitor è possibile ascoltare "in diretta" i suoni in entrata dall'altoparlante del monitor, con l'esclusione temporanea dell'immagine video. In questo modo si può regolare il volume di una sorgente audio direttamente durante l'ascolto del segnale audio.

Conclusioni

La considerazione finale è favorevole, anche il prezzo è allineato al mercato: in conclusione si tratta di un prodotto interessante e competitivo anche se con qualche neo, come già specificato, ma che potrà forse essere migliorato in una versione successiva del software di gestione. ▲

**Audio Video digitizer è in vendita presso: P.G. Elettronica s.n.c.
Via Garessio, 48/1 - Torino
Tel.011/632549**

Top Ten

a cura del servizio Lago Softmail
Via Napoleona,16 - 22100 Como
Tel. 031 - 300174

1	▲	Space Ace	Ready Soft
2	(E)	Dragon's Lair 2 (Escape from ...)	Ready Soft
3	▲	Future Wars	Delphine
4	↔	It Came from the Desert	Cinemaware
5	▼	Shadow of the Beast	Psygnosis
6	(E)	Ghouls'n' Ghosts	US Gold
7	▼	Ghostbuster 2	Activision
8	(E)	F 16 Combat Pilot	Digital Integration
9	(E)	Bomber Fighter	Activision
10	▼	Indiana Jones	Lucas Film Games

a cura di G. Biagini e A. Laus

Nota dei voti

Grafica
Tiene conto della cura nel disegno dei fondali e degli sprite, dell'animazione, dello screen dei titoli.

Sonoro
Valuta la colonna sonora e gli effetti sonori.

Giocabilità
Tiene conto della varietà del gioco, delle opzioni e la possibilità di giocare in due e di eventuali scelte.

Durata
Indica per quanto tempo ci giocherete.

Generale
E' il voto generale dei valori precedenti e di tutto ciò che accompagna il gioco: confezione, manuale, gadget, ecc.

I voti vengono espressi in percentuale da 1 a 100.

F-16 Combat Pilot

Grafica	90
Sonoro	80
Giocabilità	70
Durata	80
Generale	85

Software house: Digital Integration
Prezzo: 59.000 lire

Finalmente qualcuno ha pensato di affiancare la semplice e ormai strasfruttata simulazione di volo ad una avvincente fase preparatoria in cui si è in grado di scegliere, oltre alla missione da portare a termine, anche il tragitto da seguire, le armi da montare sul nostro F-16 e, in alcune

missioni, anche le condizioni atmosferiche. F-16 Combat Pilot è la fusione di questi due elementi: un'ottima simulazione di volo unita ad una avvincente fase preparatoria. L'elenco delle missioni è davvero lungo, ma se non vi bastassero quelle proposte, potrete, dopo averle portate a termine tutte, inventarvene di nuove, disponendo persino di una squadriglia personale a cui impartire ordini. E', inoltre, possibile, collegando due computer giocare con la funzione Gladiator, che prevede lo scontro diretto tra due avversari, in cielo. Il primo sarete voi e il secondo un vostro amico; attenti però: uno solo dei due tornerà a casa... Devo poi segnalare il manuale, in italiano, che, oltre a spiegare nei minimi dettagli gli strumenti dell'aereo, vi istruisce anche sulle varie missioni da portare a termine. C'è un capitolo dedicato alle armi di cui vi potete equipaggiare, che, tra l'altro, sono la perfetta simulazione di quelle usate dall'esercito degli Stati Uniti. Infine, si preoccupa persino di darvi alcuni suggerimenti circa le principali tattiche di attacco e di difesa. Utile, a mio parere la tabella fornita a parte, che, almeno nei primi voli, vi aiuterà molto, nell'identificazione della strumentazione e nella ricerca dei comandi. La strumentazione, esatta riproduzione di quella presente sui moderni F-16, è ben studiata graficamente; una pecca è forse l'impossibilità di vedere l'aereo dall'esterno. Nel complesso comunque, si tratta di un'ottima simulazione, anche se poco adatta ai principianti che, però, troverà molti consensi tra i "professionisti" del volo.

Distribuito da: Leader Distribuzione

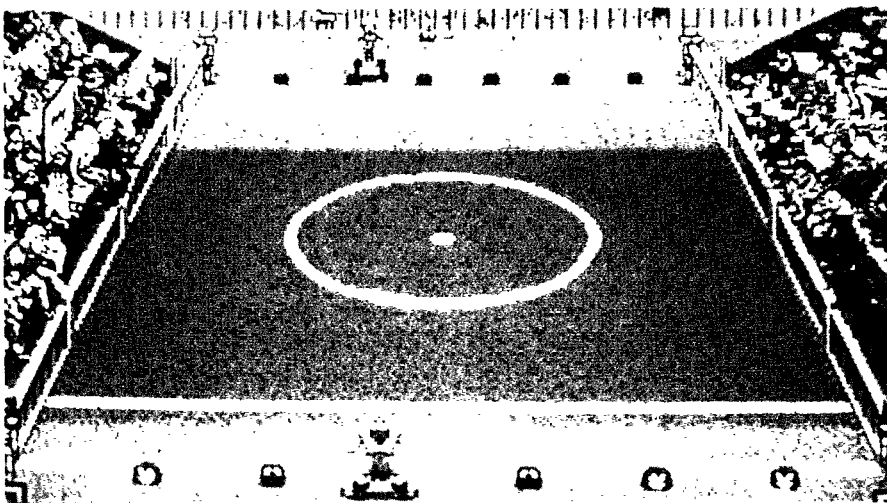
Grand Monster Slam

Grafica	90
Sonoro	78
Giocabilità	84
Durata	87
Generale	81

Software house: Rainbow Arts
Prezzo: 29.000 lire

Dalla Golden Goblins, poliedrica software house germanica meglio conosciuta come Rainbow Arts, un videogioco veramente interessante, proiettato in un futuro dove vivono mostri di tutti i tipi, questo Grand Monster Slam è una specialità sportiva molto originale. Regola fondamentale del gioco è quella di raggiungere il centrocampo, cosa non semplice dato che sia voi che il vostro avversario potete lanciarvi degli altri mostri di forma sferica completamente pelosi che prima di prendere il calcione tremano tutti.
Lo scenario del gioco è quindi un campo verde tipo quello da calcio con alcune linee in meno, i lanci dei

palla-mostri possono essere fatti con le più svariate angolazioni e generalmente è bene rivolgerli verso l'avversario che, se colpito, abbisognerà di qualche secondo per rimettersi in piedi e ricominciare a calciare. Finita questa prima parte del gioco si passa alla seconda prova: la vendetta dei Beloms (le palle-mostri) dove dovrete superare un muretto che vi arriva fino alla cinta. Attenzione agli arbitri che scendono dal cielo con una corda e attenzione a non commettere troppe infrazioni! Il gioco tecnicamente è molto vario, unisce regole senza dubbio insolite ad azioni di gioco particolarmente avvincenti; all'inizio certo ci sono dei problemi, soprattutto nel determinare l'angolazione della palla e il tipo di traiettoria parabolica che gli si deve dare. Una grande nota di merito al manuale molto chiaro e con una storia interessante e spiritosa.

Grand Monster Slam è distribuito da: Leader Distribuzione

Red Heat

Grafica	91
Sonoro	61
Giocabilità	66
Durata	68
Generale	79

Software house: Ocean
Prezzo: 39.000 lire

Ultimo lavoro della Ocean che si ispira questa volta alle gesta di Arnold Schwarzenegger, il gioco consiste nel combattere gli organizzatori di un giro di droga clandestino partendo dalla vostra patria madre, l'Unione Sovietica, per arrivare a Los Angeles, l'epicentro dello spaccio di stupefacenti.
Chi più adatto di Ivan Danko, capo della sezione omicidi della polizia di stato di Mosca? A voi quindi la guida dell'eroe che è giunto in America volando sulle ali della "Perestrojka". Tralasciando il significato politico che può avere una cooperazione tra i due blocchi, ricordiamo che quella di cui sopra è la trama del film che in Italia è uscito con il nome di DANKO. La Ocean spesso crea giochi ispirati a famose pellicole cinematografiche, ma i risultati generalmente sono più convincenti; quantunque la grafica sia curatissima e presenti degli sprite particolarmente definiti ed estesi, il gioco è poco avvincente, e a volte pecca un po' di incontrollabilità. Mi spiego: il primo schermo si svolge nella sauna di una

palestra sovietica dove si deve combattere a pugni contro un esercito di culturisti particolarmente minacciosi; niente di strano fino a che non cominciano a volare oggetti, e questo fa perdere al gioco quella veridicità che poteva avere in un primo momento. Successivamente si esce dalla sauna e si combatte sulla neve: i nemici con cui ci dobbiamo scontrare sono nudi (come quelli nella sauna) e ci tirano le palle di neve.
Per quel che riguarda la grafica, comunque, il prodotto è ben curato; all'inizio del gioco c'è un'immagine del film molto ben digitalizzata e tutta l'azione che vi si svolge comprende degli scroll disegnati con particolare cura.
Da notare anche la possibilità di prendere oggetti che serviranno nella seconda parte del gioco.

Red Heat è distribuito da:
Leader Distribuzione
Via Mazzini, 15 - Casciago (Va)
Tel. 0332/212255

Indiana Jones and the Last Crusade (adventure)

Grafica	85
Sonoro	70
Giocabilità	85
Durata	80
Generale	85

Software House: Lucas Film Games
Prezzo: 59.000 lire

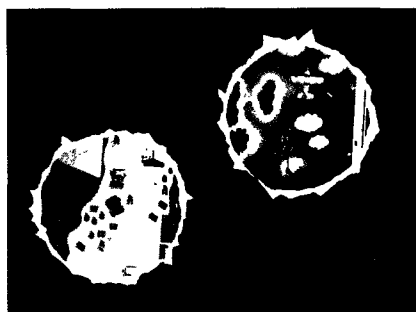
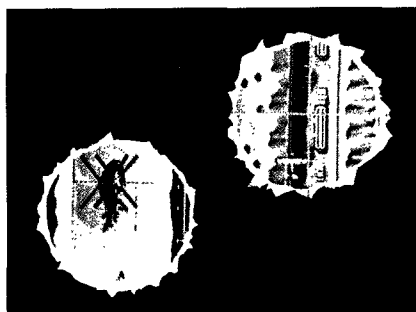
Ecco la versione "computerizzata" del film che in questa stagione ha letteralmente sbancato i botteghini. Si tratta di un adventure dalla grafica coinvolgente e con una giocabilità notevole, facilitata dal completo utilizzo del mouse, al posto dell'odiata tastiera; avete capito bene: non bisogna scrivere niente.
La trama del gioco è identica a quella del film, con il vantaggio che qui, siete voi a decidere le mosse

del nostro Indiana Jones.
Voi comandate le azioni di Indy (qualche volta anche quelle di suo padre, Henry), alla ricerca del sacro Graal. Nella maggior parte dei casi, vi ritroverete a percorrere le stesse strade che avete visto percorrere al nostro eroe nel film, ma altre volte vi troverete in luoghi sconosciuti e dovrete cavarvela da soli! Qui è la novità tanto attesa del gioco: molti schermi e situazioni nuove con cui mettersi alla prova. La confezione, oltre ai tre dischetti contenenti il programma, vi fornisce una copia del famoso diario del signor Jones, senza il quale avreste serie difficoltà a raggiungere la meta agognata. Siete provvisti anche di una speciale tabella di traduzioni che Indy dovrà usare per decifrare messaggi da antiche scritture. Il gioco, come il manuale, è tradotto in italiano e questo dovrebbe facilitarvi le cose. Una pecca è forse la lentezza dei movimenti dei personaggi che rallenta notevolmente l'azione, soprattutto se devono muoversi da una parte all'altra dello schermo.
Si tratta di un bell'adventure che, tuttavia, richiede molta pazienza; nulla a che vedere, in questo, con il suo omonimo serie arcade.
Distribuito da: Leader Distribuzione

James Bond 007 Licence to Kill

Grafica	70
Sonoro	65
Giocabilità	68
Durata	73
Generale	71

Software house: Domark
Prezzo: 29.000 lire

Tutti conoscono l'agente segreto 007, James Bond, e probabilmente tutti l'avranno voluto impersonare perlomeno in un videogioco.
La DOMARK, software inglese, ha pensato a voi.
La trama è molto semplice, ma l'azione è molto varia: dovete infatti combattere dall'elicottero, a piedi, bloccare il timone dell'aereo di Sanchez, il narcotrafficante vostro nemico, fare dello sci d'acqua, combattere contro dei sommozzatori e, infine, saltare da un camion all'altro per distruggere le partite di eroina del rivale.
Licence to Kill è un gioco graficamente ben curato, gli scroll sono molto fluidi e le animazioni buone, i colori vantano un'ottima scelta di palette ma gli sprite non sono sempre ben definiti.
A volte l'azione è molto intricata, al punto da non capire chi siamo noi: dopo un paio di partite, però, tutto cambia e se vi piacciono i giochi multidisciplinari non fatevelo scappare.
All'inizio si incontrano sicuramente dei problemi, soprattutto nel combattimento sull'elicottero sorvolando la città di Cray Cay nell'inseguire la Jeep di Sanchez, è molto difficile evitare i colpi e non si ha bene il senso dell' altezza degli edifici, per cui è molto facile scontrarsi.
Da notare che in alcune parti il gioco è potentemente autoguidato attraverso dei reticoli che ti suggeriscono cosa fare; per il resto sulla parte destra dello schermo sono presenti una serie di apparecchiature che ti aiuteranno nel gioco.
Un gioco carino per gli amanti del genere.

Licence to Kill è distribuito da:
Leader Distribuzione
Via Mazzini, 15
Casciago (Va)
Tel. 0332/212255

Dominator

Grafica	68
Sonoro	65
Giocabilità	77
Durata	60
Generale	65

Software house: System 3
Prezzo: 39.000 lire

Signore e signori sono lieto di presentarvi un gioco veramente "opaco": ricorda i primi programmi del 64, più che rispettabili per quei tempi, ma al giorno d'oggi ci sono gli AMIGA, cari amici della SYSTEM 3.
Il programma è stato sviluppato in diverse versioni per i calcolatori ad 8 bit in genere e sicuramente può essere accettabile per CPC464, C64, SPECTRUM ma per AMIGA e l'ST assolutamente no !
Vi trovate in un mondo dove sparare è necessario per sopravvivere, combattete se non volete essere distrutti. L'astronave che dovete guidare è mal controllabile e deve passare in alcuni cunicoli troppo stretti; vi è un piano di scroll che rappresenta l'unico scenario del gioco e che è copiato dal quarto schermo di XENON, che dovrebbe rappresentare la rugosa pelle di un

mostro pieno di cavilli pronti a distruggervi, solo che il tutto è animato in maniera indecente. Ci si annoia e basta. Il manuale di istruzioni ne parla come di un arcade dalle superbe qualità grafiche, ma a mio parere non è assolutamente così: ho già parlato dello scenario che non è tra i più originali, la gestione degli scroll non è tra le migliori e gli effetti sonori, se si esclude la musica iniziale, sono poco piacevoli. In conclusione la SYSTEM 3 potrebbe fare programmi migliori se venissero progettati per funzionare su AMIGA.
Distribuito da: Leader Distribuzione

Astaroth

Grafica	88
Sonoro	80
Giocabilità	86
Durata	88
Generale	85

Software house: Hewson
Prezzo: 29.000 lire

Astaroth, l'angelo della morte, una donna bellissima ma altrettanto pericolosa, con ali di fata e fascino demoniaco: quello sarà il vostro nemico.
Il programmatore dice di aver progettato Astaroth dopo aver letto un libro di Stephen King, Carrie, dove si parla di poteri soprannaturali, la chiave appunto del gioco. Voi siete un omino vestito alla Zorro, per in-

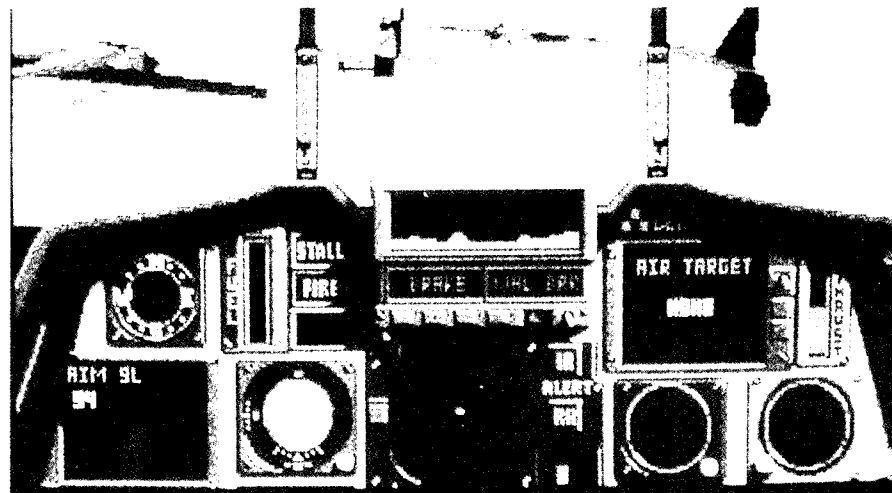
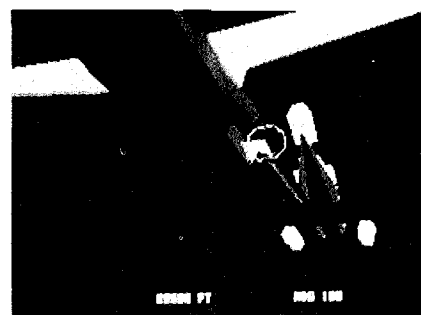
tenderci, che corre in un mondo a più livelli per cercare di distruggere Astaroth, cosa tra l'altro possibile solo dopo aver raccolto i nove poteri paranormali che si guadagnano via via nel gioco.
Inizialmente, quindi, aiutati dalle sole vostre forze, dovete farvi strada nei cunicoli dove, manco a farlo apposta, cascano gocce di acido, una serie di teschi fanno un sabba infernale, e voi naturalmente dovete cercare di evitarli per arrivare a svuotare una delle nove gerle contenenti i poteri.
I poteri sono esattamente nove: Telecinesi, la facoltà di spostare gli oggetti con il pensiero, Pirocinesi, la capacità di creare lingue di fuoco, Telepatia, il potere di comunicare con la mente, Metamorfosi, la possibilità di cambiare la forma del proprio corpo, Levitazione, abilità di muoversi su un cuscino d'aria, Trasmigrazione, abilità di spostare il proprio corpo con il pensiero, Criocinesi, possibilità di congelare gli oggetti con il pensiero, Visione notturna, per vedere anche nelle stanze buie e, infine, la Meditazione, la capacità di rilassarsi e riprendere coscienza del proprio corpo.
Il gioco, pur essendo molto semplice come grafica, è particolarmente piacevole; a volte ci sono degli effetti speciali che lasciano un po' a desiderare, però l'azione si svolge con continuità. Non garantisce grandi emozioni, però si lascia giocare piacevolmente.

Astaroth è distribuito da:
Leader Distribuzione
Via Mazzini, 15
Casciago (Va)
Tel. 0332/212255

Bomber Fighter

Grafica	70
Sonoro	65
Giocabilità	85
Durata	81
Generale	80

Software House: Activision
Prezzo: 59.000 lire

Il gioco è una simulazione delle gare di bombardamento del Comando Strategico Aereo (SAC) e l'azione si svolge nei cieli del Sud Dakota, Nord Dakota, Wyoming e Montana. Avete la possibilità di scegliere il vostro aereo tra una gamma di sette tra i più potenti velivoli militari esistenti: vi è permesso insomma di pilotare dall'F-15 Strike Eagle dell'aviazione americana, al MIG 27 sovietico. Il gioco si apre proprio con la scelta del vostro aereo e di quello del vostro avversario; in questa fase è possibile visualizzare gli aerei sia in dure dimensioni chè in tre; l'eccellente grafica usata per il disegno bi-dimensionale è stata accantonata per quello tri-dimensionale; lo stesso dicasi per la cabina del velivolo che risulta ben studiata, mentre la visione tridimensionale del vostro aereo in volo è molto carente graficamente. Prima del decollo scegliete la missione, le armi e potete avere una ricognizione via satellite del vostro obiettivo e delle zone circostanti.

La strumentazione è la stessa per tutti e sette gli aerei e questo vi facilita molto le cose; l'aereo è nel complesso facile da manovrare, anche se ci si trova un po' impacciati nella fase di combattimento.

Da qui il consiglio di fare un po' di pratica prima di distruggere un apparecchio da molti milioni di dollari. Per questo risulta utile la funzione Free-flight. Il manuale, in italiano, è piuttosto conciso: si dilunga all'inizio nel raccontare la storia della gara e non parla affatto dei comandi dell'aereo e della sua strumentazione. Fortunatamente è stata allegata una tabella che riporta tutti i comandi per fare partire il vostro mezzo.

Sempre nel manuale sono ampiamente spiegate le caratteristiche di tutti gli aerei disponibili, i vostri e quelli avversari. C'è anche una descrizione dettagliata delle armi di cui potete dotarvi. Infine, regge poco il confronto con gli altri programmi di questo tipo, ma certamente è molto giocabile e divertente, per provare l'ebbrezza del volo.

Distribuito da: Leader Distribuzione

COMPILARE E SPEDIRE IN BUSTA CHIUSA A: GRUPPO EDITORIALE JACKSON - VIA ROSELLINI 12 - 20124 MILANO

ABBONAMENTO RIVISTE JACKSON

ATTENZIONE La cedola di richiesta abbonamenti presenta un modulo speciale con una serie di domande a cui preghiamo vivamente di rispondere con precisione.
AVVERTENZA I rinnovi entreranno in vigore automaticamente a partire dal numero successivo alla data di scadenza dell'abbonamento precedente. Per motivi tecnici, i nuovi abbonamenti saranno attivati dopo 6 settimane dalla data di ricevimento della cedola di richiesta, o in data successiva nel caso di esplicita comunicazione da parte dell'abbonato.

N.B. Per abbonamenti all'estero le tariffe dovranno essere raddoppiate. Non è prevista la spedizione via aerea

❑ NUOVO ABBONAMENTO ❑ RINNOVO

Rivista	Numeri		Prezzo
❑ BIT	numeri 11	L.	53.000
❑ PC MAGAZINE	numeri 11	L.	52.000
❑ PC FLOPPY	numeri 11	L.	105.500
❑ INFORMATICA OGGI	numeri 11	L.	52.500
❑ INFORMATICA OGGI SETTIMANALE	numeri 40	L.	32.000
❑ TRASMISSIONE DATI E TELECOM.	numeri 11	L.	51.500
❑ COMPUTERGRAFICA & DTP	numeri 11	L.	51.000
❑ NTE COMPUSCUOLA	numeri 10	L.	33.000
❑ ELETTRONICA OGGI	numeri 20	L.	112.000
❑ AUTOMAZIONE OGGI	numeri 20	L.	80.000
❑ EO NEWS SETTIMANALE	numeri 40	L.	32.500
❑ STRUMENTAZIONE E MISURE OGGI	numeri 11	L.	53.500
❑ MECCANICA OGGI	numeri 11	L.	61.500
❑ STRUMENTI MUSICALI	numeri 11	L.	50.500
❑ WATT	numeri 20	L.	31.500
❑ FARE ELETTRONICA	numeri 12	L.	58.000
❑ AMIGA MAGAZINE (disk)	numeri 11	L.	123.500
❑ SUPERCOMMODORE (disk)	numeri 11	L.	110.000
❑ SUPERCOMMODORE (tape)	numeri 11	L.	75.000
❑ PC SOFTWARE 5 1/4	numeri 11	L.	106.000
❑ PC SOFTWARE 3 1/2	numeri 11	L.	132.000
❑ PC GAMES 5 1/4	numeri 11	L.	124.000
❑ PC GAMES 3 1/2	numeri 11	L.	132.500
❑ GUIDA VIDEOGIOCHI	numeri 11	L.	31.000

MODALITA' DI PAGAMENTO

Allego assegno n. ______ di L. ______
Banca ______
Ho effettuato versamento di L. ______ sul c/c postale n.11666203 intestato a Gruppo Editoriale Jackson-Milano e allego fotocopia della ricevuta.
Addebitate l'importo di L. ______ sulla carta di credito
AMERICAN EXPRESS
BANKAMERICARD
CARTA SI'
DINERS CLUB
Data di scadenza della carta di credito ______
Data ______ Firma ______

JACKSON CARD 1990

GRUPPO EDITORIALE JACKSON

Nuova, più ricca e sempre più preziosa

Ecco tutti i vantaggi che ti offre la Jackson Card 90: ● sconti esclusivi e facilitazioni negli oltre 3000 punti di vendita convenzionati ● sconto speciale del 10% sull'acquisto di libri Jackson ● invio gratuito del catalogo libri 1990 e aggiornamento sulle ultime novità editoriali ● invio gratuito della rivista Jackson Preview Magazine ● buono acquisto speciale di L. 15.000 su ordini di libri effettuati per corrispondenza.

Ottenerla è facile.........

CEDOLA DI COMMISSIONE LIBRARIA

Si inviatemi i volumi sotto elencati:
❑ All'indirizzo privato ❑ All'indirizzo di lavoro

INDIRIZZO PRIVATO

COGNOME E NOME ______
VIA E NUMERO ______
CAP ______ CITTA' ______ PROV. ______
TEL. (______) ______ ANNO DI NASCITA 19 ______
TITOLO DI STUDIO: ❑ LAUREA ❑ MEDIA SUPERIORE ❑ MEDIA INFERIORE

INDIRIZZO LAVORO

DITTA O ENTE ______
VIA E NUMERO ______
CAP ______ CITTA' ______ PROV. ______
TEL. (______) ______ TELEX/FAX ______

INDICARE CHIARAMENTE CODICI E QUANTITA' DEI VOLUMI RICHIESTI

CODICE	Q.TA	CODICE	Q.TA	CODICE	Q.TA	CODICE	Q.TA

❑ Non sono titolare della Jackson Card
❑ Sono titolare della Jackson Card '90 n° ______
e ho quindi diritto allo sconto del 10% sino al 31/12/90

Per ricevere la Jackson Card '90 l'ordine minimo è di L.100.000.

Per ogni ordine verrà addebitato l'importo di L. 4.500 quale contributo fisso di spedizione.

MODALITA' DI PAGAMENTO

❑ Contro assegno postale ❑ Assegno allegato n° ______
Banca ______
❑ Versamento su c/c postale11666203 a Voi intestato del quale allego fotocopia della ricevuta.
❑ Carta di credito ❑ American Express ❑ Visa ❑ Diner Club ❑ Carta Si
Autorizzo l'organizzazione sopra indicata ad addebitare l'importo sul mio Conto n° ______ Scadenza ______
❑ Richiedo l'emissione della fattura (formula riservata alle aziende) e comunico il numero di Partita I.V.A. ______
Data ______ Firma ______

COMPILARE E SPEDIRE IN BUSTA CHIUSA A: GRUPPO EDITORIALE JACKSON - VIA ROSELLINI 12 - 20124 MILANO

CEDOLA DI COMMISSIONE LIBRARIA

La cedola di commissione libraria presenta una serie di domande a cui preghiamo vivamente di rispondere con precisione. I dati serviranno a qualificare meglio le caratteristiche dei lettori per poter offrire un servizio informativo sulle nuove pubblicazioni e altre novità Jackson adeguato alle esigenze specifiche di ciascuno.

NUMERO DI DIPENDENTI
A ❑ da 1 a 49 C ❑ da 250 a 999
B ❑ da 50 a 249 D ❑ da 1000 in su

SETTORE AZIENDALE
- A ❑ Acquisti
- B ❑ Vendite
- C ❑ Progettazione/Ricerca e Sviluppo
- D ❑ Marketing e Comunicazione
- E ❑ Produzione
- F ❑ Amministrazione/Personale/Finanza
- G ❑ EDP
- H ❑ Altro (specificare)

ATTIVITA' PRINCIPALE DELL'AZIENDA
- P ❑ Produzione
- D ❑ Distribuzione
- S ❑ Servizi

SETTORE MERCEOLOGICO
- A ❑ Informatica
- B ❑ Automazione Industriale
- C ❑ Meccanica
- D ❑ Elettronica
- E ❑ Strumentazione
- F ❑ Elettricità e Energia
- G ❑ Trasmissione Dati e Telecomunicazioni
- H ❑ Finanza/Banche/Assicurazioni
- I ❑ Editoria/Pubblicità/Comunicazione
- L ❑ Pubblica Amministrazione Centrale/Locale
- M ❑ Consulenza
- N ❑ Istruzione (Scuola/Università)
- O ❑ Altro (specificare)

POSIZIONE
- A ❑ Alta Direzione
- B ❑ EDP o Technical Manager
- C ❑ Dirigente
- D ❑ Tecnico/Progettista
- E ❑ Consulente/Professionista
- F ❑ Docente/Formatore
- G ❑ Studente
- H ❑ Altro (specificare)

INTERESSI PRINCIPALI
- 01❑ EDP
- 02❑ Personal Computer
- 03❑ Computer Grafica e Desktop Publishing
- 04❑ Trasmissione Dati e Reti
- 05❑ Home Computer e Videogiochi
- 06❑ Automazione Industriale
- 07❑ Meccanica
- 08❑ Strumentazione Elettronica
- 09❑ Telecomunicazioni e Telefonia
- 10❑ Elettronica Professionale
- 11❑ Elettronica Hobbistica
- 12❑ Elettrotecnica e Impianti Elettrici
- 13❑ Strumenti Musicali
- 14❑ Marketing e Management
- 15❑ Broadcast/Audio e Video Professionale
- 16❑ Didattica
- 17❑ Altro (specificare)

CHE PERSONAL COMPUTER POSSIEDE/UTILIZZA
- DOS ❑ MS DOS, OS/2 e PC compatibili
- MAC ❑ Macintosh
- AMG ❑ Amiga
- C64 ❑ Commodore 64
- VAR ❑ Altro Home Computer (spec.)

...Abbonati alle riviste o acquista libri Jackson per almeno L. 100.000

Avere la Jackson Card é davvero facile: basta abbonarsi oppure acquistare libri Jackson per almento L. 100.000 compilando le cedole che si trovano in questa pagina. Col primo numero del '90 di Jackson Preview Magazine ogni titolare riceverà la Jackson Card 90, l'elenco dei negozi convenzionati e lo speciale buono acquisto.

JACKSON CARD 19..

SERVIZIO QUALIFICAZIONE ABBONATI

ABBONAMENTO GRATUITO A 40 NUMERI, A SCELTA TRA LE SEGUENTI RIVISTE
❑ EO NEWS SETTIMANALE ❑ INFORMATICA OGGI SETTIMANALE

BARRARE LA CASELLA RELATIVA ALLA RIVISTA PRESCELTA

COGNOME E NOME ____________________
VIA È NUMERO ____________________
CAP ________ CITTA' ____________ PROV ______
TEL (________) ____________ ANNO DI NASCITA 19 ______
TITOLO DI STUDIO ❑ LAUREA ❑ MEDIA SUPERIORE ❑ MEDIA INFERIORE
NUMERO JACKSON CARD GOLD ________ SILVER ________
DITTA O ENTE ____________________
VIA E NUMERO ____________________
CAP ________ CITTA' ____________ PROV ______
TEL (________) ____________ TELEX/FAX ________

NUMERO DI DIPENDENTI
A ❑ da 1 a 49 C ❑ da 250 a 999
B ❑ da 50 a 249 D ❑ da 1000 in su

SETTORE AZIENDALE
- A ❑ Acquisti
- B ❑ Vendite
- C ❑ Progettazione/Ricerca e Sviluppo
- D ❑ Marketing e Comunicazione
- E ❑ Produzione
- F ❑ Amministrazione/Personale/Finanza
- G ❑ EDP
- H ❑ Altro (specificare)

ATTIVITA' PRINCIPALE DELL'AZIENDA
- P ❑ Produzione
- D ❑ Distribuzione
- S ❑ Servizi

SETTORE MERCEOLOGICO
- A ❑ Informatica
- B ❑ Automazione Industriale
- C ❑ Meccanica
- D ❑ Elettronica
- E ❑ Strumentazione
- F ❑ Elettricità e Energia
- G ❑ Trasmissione Dati e Telecomunicazioni
- H ❑ Finanza/Banche/Assicurazioni
- I ❑ Editoria/Pubblicità/Comunicazione
- L ❑ Pubblica Amministrazione Centrale/Locale
- M ❑ Consulenza
- N ❑ Istruzione (Scuola/Università)
- O ❑ Altro (specificare)

POSIZIONE
- A ❑ Alta Direzione
- B ❑ EDP o Technical Manager
- C ❑ Dirigente
- D ❑ Tecnico/Progettista
- E ❑ Consulente/Professionista
- F ❑ Docente/Formatore
- G ❑ Studente
- H ❑ Altro (specificare)

INTERESSI PRINCIPALI
- 01❑ EDP
- 02❑ Personal Computer
- 03❑ Computer Grafica e Desktop Publishing
- 04❑ Trasmissione Dati e Reti
- 05❑ Home Computer e Videogiochi
- 06❑ Automazione Industriale
- 07❑ Meccanica
- 08❑ Strumentazione Elettronica
- 09❑ Telecomunicazioni e Telefonia
- 10❑ Elettronica Professionale
- 11❑ Elettronica Hobbistica
- 12❑ Elettrotecnica e Impianti Elettrici
- 13❑ Strumenti Musicali
- 14❑ Marketing e Management
- 15❑ Broadcast/Audio e Video Professionale
- 16❑ Didattica
- 17❑ Altro (specificare)

CHE PERSONAL COMPUTER POSSIEDE/UTILIZZA
- DOS ❑ MS DOS, OS/2 e PC compatibili
- MAC ❑ Macintosh
- AMG ❑ Amiga
- C64 ❑ Commodore 64
- VAR ❑ Altro Home Computer (spec.)

Primavera Jackson 90

1 APRILE • 30 GIUGNO

Gratis un abbonamento

Dal primo aprile al 30 giugno se acquisterai libri Jackson per un valore superiore a lire 150.000, ti sarà riservato un abbonamento omaggio per 6 mesi ad una delle seguenti riviste:

AUT. MIN.

Per ulteriori informazioni richiedi il depliant "Primavera Jackson '90" presso le migliori librerie e computer shop.

Una primavera da non perdere

Gruppo Editoriale Jackson

Hobby & Home Computer